(26)

No. 94-2003-CFX
Status: DECIDED

Title: Lotus Development Corporation, Petitioner
v.
Borland International, Inc.

Docketed:
June 7, 1995

Court: United States Court of Appeals for the First Circuit

Counsel for petitioner: Gutman,Henry Braude

Counsel for respondent: Reback,Gary L., Gehaar,Peter E., Brower,Steven

40 copies ea ptn & sep appdx.

Entry	Date	Note	Proceedings and Orders
1	Jun 7 1995	G	Petition for writ of certiorari filed.
2	Jun 7 1995		Appendix of petitioner filed.
3	Jul 6 1995		Brief of respondent Borland International, Inc. in opposition filed.
4	Jul 6 1995	G	Motion of Intellectual Property Owners for leave to file a brief as amicus curiae filed.
5	Jul 7 1995	G	Motion of Information Technology Industry Council for leave to file a brief as amicus curiae filed.
6	Jul 12 1995		DISTRIBUTED. September 26, 1995 (Page 41)
7	Jul 18 1995	X	Reply brief of petitioner filed.
8	Sep 27 1995		Motion of Intellectual Property Owners for leave to file a brief as amicus curiae GRANTED. Justice Stevens OUT.
9	Sep 27 1995		Motion of Information Technology Industry Council for leave to file a brief as amicus curiae GRANTED. Justice Stevens OUT.
10	Sep 27 1995		Petition GRANTED. The brief of petitioner is to be filed with the Clerk and served upon opposing counsel on or before 3 p.m., Thursday, November 9, 1995. The brief of respondent is to be filed with the Clerk and served upon opposing counsel on or before 3 p.m., Friday, December 8, 1995. A reply brief, if any, is to be filed with the Clerk and served upon opposing counsel on or before 3 p.m., Thursday, December 28, 1995. Rule 29.2 does not apply. Justice Stevens OUT.
			**********
12	Oct 27 1995		Record filed.
		*	Partial record proceedings United States Court of Appeals for the First Circuit (BOX) (SEALED ENVELOPE)
38	Oct 27 1995		Record filed.
		*	Certified record proceedings United States Court of Appeals for the First Circuit. (BOX)
11	Nov 1 1995		SET FOR ARGUMENT MONDAY, JANUARY 8, 1996. (3RD CASE).
13	Nov 9 1995		Joint appendix filed.
		*	Joint appendix in two volumes
14	Nov 9 1995		Brief of petitioner Lotus Development Corporation filed.
16	Nov 9 1995	G	Motion of American Intellectual Property Law Association for leave to file a brief as amicus curiae filed.
17	Nov 9 1995		CIRCULATED.
19	Nov 9 1995	X	Brief amici curiae of Digital Equipment Corporation, et al.

2 PP

No. 94-2003-CFX

Entry	Date	Note	Proceedings and Orders
			filed.
18	Nov 13 1995		LODGING consisting of twelve copies of Selden's Condensed Ledger, Reports and Condensed Memorandum Book (5th ed.1861) submitted by Counsel for the Petitioner.
20	Dec 4 1995		Motion of American Intellectual Property Law Association for leave to file a brief as amicus curiae GRANTED.
22	Dec 7 1995	G	Motion of Howard C. Anawalt for leave to file a brief as amicus curiae filed.
25	Dec 7 1995	G	Motion of Software Protection Comm. of MN Intellectual Property Law for leave to file a brief as amicus curiae filed.
35	Dec 7 1995		Record filed.
		*	Original record proceedings United States District Court for the District of Massachusetts (27 BOXES).
37	Dec 7 1995	X	Brief amicus curiae of Computer Scientists filed.
23	Dec 8 1995	X	Brief of respondent Borland International, Inc. filed.
24	Dec 8 1995		LODGING consisting of twelve copies of two videotapes submitted by counsel for the respondent.
26	Dec 8 1995	X	Brief amici curiae of Software Industry Coalition, et al. filed.
27	Dec 8 1995	X	Brief amicus curiae of Economics Professors and Scholars filed.
28	Dec 8 1995	X	Brief amicus curiae of Users Groups filed.
29	Dec 8 1995	X	Brief amicus curiae of Professors Peter S. Menell and Dennis S. Karjala filed.
30	Dec 8 1995	X	Brief amicus curiae of Software Forum filed.
31	Dec 8 1995	X	Brief amicus curiae of League for Programming Freedom filed.
32	Dec 8 1995	X	Brief amici curiae of American Committee for Interoperable Systems, et al. filed.
33	Dec 8 1995	X	Brief amicus curiae of Copyright Law Professors filed.
34	Dec 8 1995	X	Brief amicus curiae of Altai, Inc. filed.
36	Dec 15 1995		LODGING by petitioner. Twelve copies of videotape filed in the United States District Court for the District of Massachusetts (Docket No. 332, 2-35, 2-36; Docket No. 333, 3-26:4-11).
39	Dec 28 1995	X	Reply brief of petitioner Lotus Development Corporation filed.
40	Jan 8 1996		Motion of Howard C. Anawalt for leave to file a brief as amicus curiae GRANTED. Justice Stevens OUT.
41	Jan 8 1996		Motion of Software Protection Comm. of MN Intellectual Property Law for leave to file a brief as amicus curiae GRANTED. Justice Stevens OUT.
42	Jan 8 1996		ARGUED.
43	Jan 16 1996		Adjudged to be AFFIRMED. The judgment of the United States Court of Appeals for the First Circuit is affirmed by an equally divided Court. Justice Stevens took no part in the consideration or decision of this case. Opinion per curiam.

No. _____ Supreme Court, U.S.
FILED
942003 JUN 7 - 1995
OFFICE OF THE CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

PETITION FOR A WRIT OF CERTIORARI

CHARLES FRIED
1575 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4636

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

Counsel for Petitioner

QUESTION PRESENTED

Whether a computer program's particular menu command hierarchy, which the district court found to contain expression separable from its underlying idea and the functionality it describes, may be protected by copyright in light of the explicit Congressional extension of copyright to computer programs under the same principles applicable to other literary works; or whether, as the First Circuit held, Section 102(b) of the Copyright Act bars protection for any such menu command hierarchy despite its expressive characteristics, because it assists users in communicating with a computer program in order to perform useful operations.

RULE 29.1 STATEMENT

Petitioner Lotus Development Corporation has no parent corporation and no subsidiaries that are not wholly owned, except for certain foreign subsidiaries in which a minimal amount of shares (fewer than 1%), which are not publicly traded, are held by foreign nationals in accordance with local law.

TABLE OF CONTENTS

	PAGE
QUESTION PRESENTED	i
RULE 29.1 STATEMENT	ii
TABLE OF AUTHORITIES	iv
OPINIONS BELOW	1
JURISDICTION	1
CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED	1
STATEMENT OF THE CASE	1
A. Factual Background	2
B. Legal Background	7
C. Prior Proceedings in This Case	17
1. In the District Court	17
2. In the Court of Appeals	19
REASONS FOR GRANTING THE PETITION	21
I. THE DECISION OF THE COURT OF APPEALS CONFLICTS WITH DECISIONS OF NO FEWER THAN FIVE OTHER CIRCUITS	23
II. THE QUESTION OF FEDERAL LAW DECIDED BY THE COURT OF APPEALS UNSETTLES THE LAW IN AN IMPORTANT AREA AND THEREFORE THE PUBLIC INTEREST REQUIRES THIS COURT'S REVIEW	26
CONCLUSION	30

TABLE OF AUTHORITIES

Cases PAGE

Apple Computer, Inc. v. Formula Int'l Inc., 725 F.2d 521 (9th Cir. 1984) 13n

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) 12-13, 26, 29

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, __ U.S. __, 115 S. Ct. 1176 (1995) 25

Atari Games Corp. v. Oman, 888 F.2d 878 (D.C. Cir. 1989) 13

Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982) 13

Autoskill, Inc. v. National Educational Support Systems, Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, __ U.S. __, 114 S. Ct. 307 (1993) 14n, 24, 24n

Baker v. Selden, 101 U.S. 99 (1879) 9, 10, 20

Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), *cert. denied sub nom. BB Asset Mgmt. Inc. v. Symantec Corp.*, __ U.S. __, 113 S. Ct. 198 (1992) 25

Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 10, 14n, 15-16, 20n, 25, 26

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994) 14n, 16-17, 22, 23-24

PAGE

Feist Publications, Inc. v. Rural Telephone Serv. Co., 449 U.S. 340 (1991) 8-9

Gates Rubber Co. v. Bando Chemical Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) 14n, 16, 24n

Harper & Row, Publishers, Inc. v. Nation Enterprises, 471 U.S. 539 (1985) 21

Hartfield v. Peterson, 91 F.2d 998 (2d Cir. 1937) 12n

Johnson Controls, Inc. v. Phoenix Control Systems, Inc., 886 F.2d 1173 (9th Cir. 1989) 14n, 25

Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527 (5th Cir.), *cert. denied*, __ U.S. __, 115 S. Ct. 82 (1994) 14n

Mazer v. Stein, 347 U.S. 201 (1954) 12, 12n, 18, 20n, 21, 30

Morrissey v. Procter & Gamble Co., 379 F.2d 675 (1st Cir. 1967) 10

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931) 10, 15, 25

Orgel v. Clark Boardman Co., 301 F.2d 119 (2d Cir.), *cert. denied*, 371 U.S. 817 (1962) 12n

Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990) ..... *passim*

Peter Pan Fabrics v. Martin Weiner Corp., 274 F.2d 487 (2d Cir. 1960) 28n

Reiss v. National Quotation Bureau, 276 F. 717 (S.D.N.Y. 1921) 12n

PAGE

Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417 (1984) 21

Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 13-14, 16, 26

Constitutional Provision, Statutes and Public Laws

United States Constitution, art. I, § 8, cl. 8..... 1, 7

17 U.S.C. § 101 ..... 1, 3, 8, 18n, 27

§ 102(a) ..... *passim*

§ 102(b) ..... *passim*

§ 103 ..... 1, 14

§ 117 ..... 1, 27n

28 U.S.C. § 1254 1

Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124 (repealed 1831) 7

Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1873 (1974) 11n

Pub. L. No. 96-517, § 12, 94 Stat. 3015 (1980) (codified at 17 U.S.C. §§ 101, 117)..... 11n

Legislative Materials

H.R. Rep. No. 1476, 94th Cong., 2d Sess., *reprinted in* 1976 U.S. Code Cong. & Admin. News 5659 ..... 8, 11, 12, 27

H.R. Rep. No. 1307, 96th Cong., 2d Sess., *reprinted in* 1980 U.S. Code Cong. & Admin. News 6460 .... 11n

PAGE

Other Authorities

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978)..... 11, 12, 21

David Nimmer & Melville B. Nimmer, 3 *Nimmer on Copyright*, § 13.03[F][1]..... 16

Lotus Development Corporation ("Lotus") respectfully petitions for a writ of certiorari to review the judgment of the United States Court of Appeals for the First Circuit in this case.

OPINIONS BELOW

The opinion of the court of appeals (Pet. App. 1a-28a) is reported at 49 F.3d 807. The opinions of the District Court for the District of Massachusetts (Robert E. Keeton, J.) are reported at 788 F. Supp. 78 (Pet. App. 145a-182a); 799 F. Supp. 203 (Pet. App. 106a-144a); 831 F. Supp. 202 (Pet. App. 71a-105a); and 831 F. Supp. 223 (Pet. App. 29a-70a). The opinion of the district court in the related case of *Lotus Development Corp. v. Paperback Software International* is reported at 740 F. Supp. 37 (Pet. App. 183a-269a).

JURISDICTION

The Court of Appeals entered judgment on March 9, 1995. Pet. App. 1a. The jurisdiction of this Court is invoked under 28 U.S.C. § 1254.

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED

United States Constitution, art. I, § 8, cl. 8

> The Congress shall have Power . . . [t]o promote the Progress of Science and the useful Arts, by securing for limited Times to Authors . . . the exclusive Right to their . . . Writings. . . .

17 U.S.C. § 101, 17 U.S.C. § 102, 17 U.S.C. § 103, 17 U.S.C. § 117. (The full text of the statutory provisions involved is set forth in the accompanying Appendix (Pet. App. at 270a-280a).)

STATEMENT OF THE CASE

This case presents issues of fundamental importance concerning the scope of copyright protection for computer software and the application of Section 102(b) of the Copyright Act, 17 U.S.C. § 102(b), to this type of literary work. In a series of decisions in this and a related case (Pet. App. 29a-269a), the

district court (Keeton, J.) developed criteria for applying to computer programs the "idea/expression" distinction embodied in Section 102(b) and traditionally employed to delineate the scope of copyright protection for literary works. The district court's analysis, which has become virtually canonical, has been cited with approval by the Tenth, Ninth, and Second Circuits. Its conclusion—that the menu command hierarchy of petitioner's Lotus 1-2-3 program contains protectable expression—has been expressly relied upon by the Fifth Circuit.

The First Circuit, openly acknowledging the conflict between its analysis and that of other circuits (Pet. App. at 21a-22a), reversed the district court and adopted an interpretation of Section 102(b) that reads a new threshold requirement into the copyrightability provision in Section 102(a). The First Circuit rejected the application of the "idea/expression" dichotomy to the 1-2-3 menu command hierarchy and concluded that it is barred from copyright protection, despite any original expression it may contain, because it can be characterized as comprising part of a "method of operation." Pet. App. at 21a. In so ruling, the court below denied copyright protection for the element of the 1-2-3 computer program that communicates most directly with the user, which the district court found served an explanatory and informative purpose separable from its ultimate functional uses. Because all computer programs state a "method of operation," or "method for operating the computer" (*id.* at 20a), the First Circuit's decision has unsettled the law of copyright as applied to many elements of computer programs, not just their menu command hierarchies. The practical needs for clarification and for national uniformity of law justify this Court's review.

A. Factual Background

The district court's opinion in *Lotus Development Corp. v. Paperback Software Int'l* (Pet. App. at 184a-193a & 226a-229a) (hereinafter "*Paperback*") contains a useful primer on the nature of computer programs and the elements of the programs at issue. We draw from that analysis and summarize its key points.

For Copyright Act purposes, Congress defined a "computer program" as follows:

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result. (17 U.S.C. § 101.)

The principal written manifestation of a computer program is its source or object code. Pet. App. at 187a. Source code typically is written in a form of programming language, such as BASIC or FORTRAN, that uses complex symbolic names and rules of syntax to express instructions for the computer to execute. *Id.* at 188a. The source code must be converted or "compiled" into a binary form that the digital computer can recognize, called a machine language, before the computer can execute the program. *Id.* at 189a. Compiled source code is called "object code." *Id.*

The user usually never sees or is even aware of the object or source code during operation of the program. The portion of the program the user communicates with is called the "user interface." *Id.* at 226a. This normally consists of a series of textual messages or visual images displayed on the video monitor that the program causes the computer to generate in response to instructions from the user or as a result of preceding operations. The user interface also includes the messages the user communicates to the program, which the program interprets, in context, as an instruction to the computer. Because a program's user interface is perceived in a form that usually is distinct from the program code that generates it, user interfaces often have been referred to as "non-literal elements" of computer programs. *Id.* at 258a-260a. User interfaces are, however, generated by and represented in the program code (*id.*), and frequently represent a substantial part of both the creative design effort and value in a program. *Id.* at 135a, 212a, 235a & 250a.

From a commercial perspective, personal computer programs generally are divided into two categories: operating system programs and application programs. *Id.* at 186a. Operating system programs control the basic functions of the computer, such as the internal allocation of computer memory. *Id.* Application programs permit a user to perform a set of related operations directed towards particular tasks, such as word processing or

database management. *Id.* Typically, application programs are designed to work with a particular operating system, for example, DOS or UNIX, and will not work with a computer directly. *Id.* Lotus 1-2-3 is an example of a type of application program known as a "spreadsheet" program. *id.* These provide users with the ability to perform arithmetical and mathematical operations on numerical data entered by the user in an electronic representation of a spreadsheet grid, arranged in columns and rows; to reconfigure the style or layout of the spreadsheet to suit the user's particular needs; and to manipulate and present the data in the spreadsheet in a variety of ways, such as different tabular or graphical display formats. *Id.* at 226a-227a.

In order to cause the 1-2-3 program to perform a functional operation, the user first must communicate an instruction to the program that the program will recognize. Like a short-form instruction manual or reference guide, the Lotus 1-2-3 menu command hierarchy provides information to assist users in selecting the appropriate instructions that will cause the program to perform certain types of tasks.[1] *Id.* at 78a-79a. When the user presses the "/" (slash) key, the program displays a "menu" of ten words representing the array of available "menu commands" or instructions that the program is designed to recognize at that point, beginning with "WORKSHEET," "RANGE," "COPY," and "MOVE."[2] *Id.* at 227a-228a. Most of these "commands" do not cause the program to perform any operation other than to display another menu, or sub-menu, of further choices, and so on until a particular instruction to the program is specified.[3] At each menu level, the user indicates a selection by moving the cursor key to highlight an item and striking the "Enter" key, or by striking a letter key corresponding to the first letter of the word representing the desired menu command, *e.g.*, "W" for "WORKSHEET." The ordering and arrangement of the various menus and dependent submenus is called the "menu tree" or "menu command hierarchy." *Id.* at 79a-81a. There are 469 menu commands in all, arranged in more than 50 different menus. *Id.* at 4a & 92a.

[1] These do not, however, include arithmetical and mathematical calculations, which a user indicates by operands and other notations entered in the spreadsheet itself. *Id.* at 232a-233a.

[2] The description in text is of the versions of Lotus 1-2-3 that Borland copied, all of which were published prior to April 1986. *Id.* at 81a, 96a & 113a. Lotus subsequently released numerous revised versions of Lotus 1-2-3 containing additional user interface features.

[3] For example, the "WORKSHEET" command leads to a submenu of nine new choices, beginning with "GLOBAL."

The 1-2-3 menu command hierarchy was designed in 1982 by Lotus' founder, Mitchell Kapor, and a team working under his direction. *Id.* at 231a. Kapor and his team first specified the selection of functional capabilities that the program would provide (such as changing the widths of individual columns in the spreadsheet), and implemented those capabilities in the program's code. *Id.* at 288a. The menu commands reflected Kapor's attempt to express those capabilities to users in words "that would intelligently convey to the user the purpose of each command." *Id.* at 291a. The hierarchical arrangement of the menus was intended to "reflect a structured approach that communicated the product's underlying functionality." *Id.* Kapor and his team spent hundreds of hours refining the choice of each word to use in the menus, the order in which those words would appear within each menu, and the organization of the menus in a hierarchical structure. *Id.* The menu hierarchy he ultimately selected "was based largely on my intuition and subjective judgment . . . trying as best I could to imagine myself in the role of a typical user."[4] *Id.* at 292a.

As the district court found in this action, the 1-2-3 menu command hierarchy is only one of a very large number of possible ways to express to users the universe of available instructions for performing operations in a spreadsheet program. *Id.* at 86a-88a & 131a-133a. The record before the district court con-

[4] 1-2-3 also was designed to allow the user to create "macros," or sequences of frequently-used commands that can be executed with a single keystroke. *Id.* at 290a. Macros may include commands other than menu commands. *Id.* at 228a-229a. In writing a macro, the user indicates a menu command either by referring to its first letter ("C" for "COPY") or by writing a series of positional commands corresponding to the movement of the cursor followed by the "ENTER" command, *i.e.*, "{RIGHT}{RIGHT}{ENTER}." *Id.* at 31a-33a.

tained numerous examples of contemporaneous programs that performed the same basic spreadsheet functions, but employed different menu hierarchies and menu commands. *Id.* at 88a. Indeed, the district court found that, even if a software developer voluntarily restricted itself to designing a program that provided *exactly* the same set of functional capabilities that 1-2-3 provided, using the same type of hierarchical menu structure, there would still be millions of possible permutations for an acceptable spreadsheet menu command hierarchy. *Id.* at 131a. Other words could be chosen to represent each menu command; the order of the words within each menu could be altered; and the placement of commands in particular menus—*i.e.*, the decisions as to which commands should depend upon other commands, at what depth in the hierarchical arrangement—could be varied. *Id.* at 131a-132a. It is the copyrightability of the *entire* 1-2-3 menu command hierarchy, taken as a whole, and *not* the protection of any individual word viewed in isolation, such as the relatively standard menu commands "PRINT" or "QUIT," that is at issue in this case. *Id.* at 86a-89a.

Following its introduction in January 1983, Lotus 1-2-3 achieved great commercial success and became the most popular spreadsheet program for use with DOS operating systems. *Id.* at 230a-231a. Lotus' success was rapidly followed by imitators such as Paperback Software, who sought to enter the spreadsheet market with self-described 1-2-3 "clones," or programs displaying menu command hierarchies deliberately copied from Lotus 1-2-3, which sold at substantially lower prices. *Id.* at 147a-148a & 236a-238a. Lotus commenced the *Paperback* case in January 1987, alleging that copying of the 1-2-3 user interface was a violation of its copyrights in 1-2-3. *Id.* at 96a. With full awareness of that case, Borland released the first version of its spreadsheet products, called "Quattro," in November 1987, and a second product, called "Quattro Pro," in November 1989. *Id.* at 96a & 100a.

Quattro and Quattro Pro differed from earlier "clones" chiefly by providing alternate menu trees that the user could select: a so-called "native" menu tree, which was developed independently by Borland and was materially different both in menu

content and hierarchical arrangement from 1-2-3; and a "1-2-3 emulation" menu tree, which contained a virtually identical copy of the Lotus 1-2-3 menu command hierarchy. *Id.* at 82a. Both types of menu tree allowed users to invoke the same set of functional capabilities. *Id.* at 131a. Like the earlier "clone" makers, Borland copied the 1-2-3 menu command hierarchy for commercial reasons, in order to try to persuade 1-2-3 users to switch to its products by promoting the presence of the "familiar" 1-2-3 menus. *Id.* at 61a-62a. Every version of Borland's spreadsheet products sold before the district court's ruling in this case contained, in some form, a virtually identical copy of the 1-2-3 menu command hierarchy.[5] *Id.* at 33a & 82a.

B. Legal Background

Congress' authority to confer copyright protection is stated in the Constitution of the United States, art. I, § 8, cl. 8:

> The Congress shall have Power . . . [t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors . . . the exclusive Right to their . . . Writings . . . .

The First Congress applied this power to protect useful and utilitarian writings, as well as artistic and aesthetic works, extending copyright protection to "any map, chart, book or books already printed." Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124, 124 (repealed 1831).

When Congress adopted the current Copyright Act in 1976, it deliberately defined the "works of authorship" eligible for copyright protection in an open-ended and general manner. Thus, Section 102(a) provides that:

---

[5] After Lotus commenced this action, Borland introduced a new feature in subsequent versions of Quattro Pro called the "Key Reader." *Id.* at 33a. This was a modification of the "1-2-3 emulation" menu tree that was stripped down to just the first letters of the 1-2-3 menu commands and hidden inside the program, in what Borland referred to in its own internal documents as "phantom" 1-2-3 menus. *Id.* at 35a-36a. Although the Key Reader would not display these "phantom" menus to the user, it allowed users to execute 1-2-3 macros while working with the "native" Quattro Pro menus. *Id.* at 33a.

> Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression, *now known or later developed*, from which they can be perceived, reproduced or otherwise communicated, either directly or with the aid of a machine or device. (17 U.S.C. § 102(a) (emphasis supplied).)

As the emphasized language indicates, Congress intended that the types of protected "works" were not to be restricted to those expressed in media then in existence or in popular use, but also would embrace new forms of expression that would become possible only through future technological advances.

Section 102(a) goes on to state an "illustrative and not limitative" list of seven categories of protected "works of authorship," the first of which is "literary works." 17 U.S.C. § 102(a). Literary works are defined as works, "other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects . . . in which they are embodied." 17 U.S.C. § 101.[6] Congress specifically intended that computer programs be treated as "literary works":

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. *It also includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.* (H.R. Rep. No. 1476, 94th Cong., 2d Sess. at 54 (the "House Report"), *reprinted in* 1976 U.S. Code Cong. & Admin. News 5659, 5667 (emphasis supplied).)

The fact that a work is eligible for copyright, however, does not mean that every element of the work may be protected against copying by others, as this Court observed in *Feist Publications, Inc. v. Rural Telephone Serv. Co.*, 449 U.S. 340, 348 (1991). The *scope* of protection afforded to a copyrighted work is subject to important limiting principles, derived from the

[6] Both a computer program's code and its menu command hierarchy are, therefore, expressions of a literary work.

Constitution itself. The first of these is that copyright will protect only so much of a work as is original to, or created by, the author. For example, copyright in a factual compilation will extend only to the original selection, coordination, or arrangement of the facts contributed by the author, but not to the facts themselves, which the author did not create. *Id.* at 348. Relatedly, because the purpose of copyright is "[t]o promote the Progress of Science and the useful Arts," "copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work." *Id.* at 349-50.

This distinction between an idea, which copyright will not protect, and a particular expression of that idea, which may be protected if original to the author, "applies to all works of authorship." *Id.* at 350. Its origin lies in this Court's seminal decision in *Baker v. Selden*, 101 U.S. 99 (1879). Selden published a book describing a method of double-entry accounting and sued Baker for infringement when Baker subsequently published another work describing the same method in his own words. Concluding that Baker had not infringed Selden's copyright, this Court explained:

> Where the art [a book] teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public; not given for the purpose of publication in other works explanatory of the art, but for the purpose of practical application . . . . The use by another of the same methods of statement, whether in words or illustrations, in a book published for teaching the art, would undoubtedly be an infringement of the copyright. (101 U.S. at 103-104.)

Baker v. Selden always has been understood to permit the free use of the ideas or methods explained in a useful literary work, while prohibiting another from copying an author's particular description or manner of articulation of those ideas or methods—*i.e.*, their expression.

In the years between *Baker v. Selden* and the passage of the 1976 Copyright Act, the courts recognized that copyright infringement could occur although the infringer had not copied the literal text of a work, or "else a plagiarist would escape by immaterial variations." *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930) (L. Hand, J.), *cert. denied*, 282 U.S. 902 (1931). To guide the application of the idea/expression distinction to such "non-literal" elements of copyrighted works, Judge Hand devised the "abstractions" test, articulated in *Nichols*:

> Upon any work . . . a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the [work] is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the [author] could prevent the use of his "ideas," to which, apart from their expression, his property is never extended. (45 F.2d at 121.)

As the courts attempted to draw the boundary between idea and expression, they developed the concept of "merger." This doctrine, too, has its roots in *Baker v. Selden*. *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 707 (2d Cir. 1992). If a particular form of expression is necessary to use of an idea—such as "the methods and diagrams" that were "necessary incidents" to the use of Baker's bookkeeping system (101 U.S. at 103)—then the expression is said to "merge" with the idea. *Altai*, 982 F.2d at 704. Under the merger doctrine, when an idea is capable of only one or a limited number of forms of expression, so that copyright in those few forms effectively would preclude others from using the idea, the expression is uncopyrightable. *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967).

Congress codified the judicially developed idea/expression distinction in Section 102(b) of the 1976 Copyright Act, which provides:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work. (17 U.S.C. § 102(b).)

As the legislative history explains:

> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged. (House Report at 57, *reprinted in* 1976 U.S. Code Cong. & Admin. News at 5670.)

The legislative history makes clear that Congress specifically intended the courts to apply this distinction to delineate the scope of protection for computer programs. *Id.*

In 1980, Congress reaffirmed this intention when it adopted, without material modification, the recommendations of the National Commission on New Technological Uses of Copyrighted Works ("CONTU").[7] Congress had established CONTU in 1974 for the purpose, *inter alia*, of studying the relationship between computers and copyrights, and reporting recommendations concerning what, if any, changes in the law were appropriate to accommodate the inclusion of computer programs.[8] Following extensive study, public hearings and debate, CONTU recommended that computer programs continue to receive protection as "literary works" under traditional copyright principles. *See* Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978), at 1-2 & 37-46 (hereinafter "CONTU Report").

CONTU was well aware that computer programs were inherently functional and utilitarian as well as expressive. *Id.* at 21-23. That a work can be put to useful or functional purposes, however, does not serve to bar copyright protection for the

---

[7] Pub. L. No. 96-517, § 12, 94 Stat. 3015, 3028 (1980) (codified at 17 U.S.C. §§ 101, 117). *See* H.R. Rep. No. 1307, 96th Cong., 2d Sess. at 23, *reprinted in* 1980 U.S. Code Cong. & Admin. News 6460, 6482 (the pertinent section of the bill "embodies the recommendations of [CONTU] with respect to clarifying the law of copyright of computer software").

[8] Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1873-74 (1974).

expressive elements of the work. In *Mazer v. Stein*, 347 U.S. 201, 218 (1954), this Court held that the sculptural base for a lamp was copyrightable to the extent that its expression could be separated from its utilitarian function, declaring that the "intended use or use in industry of an article eligible for copyright" did not preclude protection for such a work.[9] Copyright has long protected such useful forms of writing as maps and charts, dictionaries, catalogs, textbooks, law reporters, and code books,[10] as the legislative history of the 1976 Act recognized. House Report at 53-54, *reprinted in* 1976 U.S. Code Cong. & Admin. News at 5666-67.

CONTU further understood that computer programs potentially could be characterized as part of a "process" or "method of operation" under Section 102(b). CONTU Report at 18-20. Nevertheless, CONTU expressed a firm belief that the "idea/expression" distinction would suffice to guide the courts to draw an appropriate line that would still provide meaningful copyright protection for computer programs. As it declared:

> That the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability. (*Id.* at 21.)

This precise issue was presented in *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984). Franklin argued, as CONTU had anticipated, that Section 102(b) precluded copyright protection for the operating system program designed to work with the then-industry standard Apple II personal computer, on the ground that it was an uncopyrightable "process," "system," or "method of operation." 714 F.2d at 1250-52. Applying the

[9] This Court also held in *Mazer* that the patent and copyright laws are not mutually exclusive, and that the availability of patent protection for certain aspects of an article did not bar the application of copyright to protect other aspects. *Id.* at 217.

[10] *See, e.g., Orgel v. Clark Boardman Co.*, 301 F.2d 119 (2d Cir.), *cert. denied*, 371 U.S. 817 (1962) (upholding the copyrightability of analysis, organization, phrasing, and citation in textbook); *Hartfield v. Peterson*, 91 F.2d 998 (2d Cir. 1937) (A. Hand, J.) (holding telegram and cable code protectable); *Reiss v. National Quotation Bureau*, 276 F. 717 (S.D.N.Y. 1921) (L. Hand, J.) (same).

idea/expression distinction and merger analysis to determine that other methods existed to create a computer program that would operate the computer, the court held that the particular set of instructions Apple had created to express that idea was protected by copyright.[11] The court also rejected, as a "commercial and competitive objective" that was irrelevant to its copyrightability analysis, Franklin's argument that its copying was necessary to achieve "compatibility," or to create an operating system program that would work with the body of existing application programs designed for the Apple II system. *Id.* at 1253.

Other early decisions applied the idea/expression distinction in the context of videogames, which are a type of computer program. For example, in *Atari, Inc. v. North American Philips Consumer Electronics Corp.*, 672 F.2d 607, 615-18 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982), the Seventh Circuit applied Judge Hand's *Nichols* "abstractions" test to determine the scope of protection in plaintiff's PAC-MAN videogame. Although the court held that the game rules comprised an unprotectable "idea," it went on to examine plaintiff's implementation of that game to determine whether "the particular form in which it is expressed" provided "something 'new or additional over the idea'." 672 F.2d at 617 (citation omitted). Finding expressive elements that the "game as such" did "not dictate," the court held that defendant's "virtually identical" copying of those elements was an infringement. *Id.* at 618. *See also Atari Games Corp. v. Oman*, 888 F.2d 878, 884-86 (D.C. Cir. 1989) (R.B. Ginsburg, J.) (reversing and remanding judgment affirming refusal of the Register of Copyrights to register copyright in the BREAKOUT videogame, for failure to explain how, in the Register's view, the work was "dictated by 'functional requirements'" and did not contain expression "separable from the game itself" that would "qualify as copyrightable subject matter").

In *Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987),

[11] A year later, the Ninth Circuit adopted the Third Circuit's analysis to reach the same conclusion in *Apple Computer, Inc. v. Formula Int'l Inc.*, 725 F.2d 521, 523-25 (9th Cir. 1984).

the Third Circuit addressed the question whether copyright in a computer program extended to its non-literal elements, as is true for other forms of literary works. The defendant in *Whelan* had not copied the literal text of plaintiff's program code, but rather had comprehensively copied its structure, sequence, and organization to create a competing program to perform the same functions. Finding support for its conclusion in Section 103 of the Copyright Act, 17 U.S.C. § 103, which extends protection to "compilations" of "pre-existing materials or of data that are selected, coordinated, or arranged" in an original way (797 F.2d at 1239), the court found no reason to treat computer programs differently from other literary works in this regard. Accordingly, the court allowed "copyrightable protection beyond the literal computer code" (*id.* at 1237), extending it also to "the particular means chosen" by the programmer to achieve the program's purpose or function so long as the means are "not necessary to that purpose or function." *Id.* at 1236.

The core of the *Whelan* holding—that copyright protection extends to the non-literal elements of computer programs—has been accepted by every court of appeals to have considered the issue.[12] The circuits also have agreed that Section 102(b) should be interpreted as embodying the "idea/expression" distinction with regard to all types of utilitarian literary works, including computer programs. Although, in the years following *Whelan*, the courts sometimes have struggled to formulate an appropriate test or methodology to follow in applying this distinction to the non-literal elements of computer programs, until the First Circuit's decision in this case the courts of appeals were moving towards a consensus on this point as well.

---

[12] *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1341 (5th Cir. 1994); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 536 (5th Cir.), *cert. denied*, ___ U.S. ___, 115 S. Ct. 82 (1994); *Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, 9 F.3d 823, 840 (10th Cir. 1993); *Autoskill, Inc. v. National Educational Support Systems, Inc.*, 994 F.2d 1476, 1495 n.23 (10th Cir.), *cert. denied*, ___ U.S. ___, 114 S. Ct. 307 (1993); *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir. 1992); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

The district court's decision in *Paperback* has played a prominent role in this evolution. To determine whether the 1-2-3 user interface contained copyrightable expression, the district court developed a three-part test, taking Judge Hand's "abstractions" test from *Nichols v. Universal Pictures*, 45 F.2d at 121, as its starting point. In the first step, the court seeks to identify various conceptions of the "idea" behind the elements of the work at issue, ranging from the most generalized (an electronic spreadsheet), to the most particularized (the precise contents and arrangement of the 1-2-3 menu command hierarchy). Pet. App. at 220a-222a. In the second step, the court examines each element of the program tentatively identified as expression to determine whether it is "essential to" or merges with the idea itself, or is one of only a few ways to express the idea. *Id.* at 222a. In the third step, the court considers any non-essential elements of expression that remain after the second step, to ascertain whether they are qualitatively substantial enough to make their appropriation unlawful. *Id.*

Applying this test to the 1-2-3 user interface following a twelve-day bench trial, the district court concluded that certain elements were unprotected because they were essential to every expression of an electronic spreadsheet, were obvious, or were standard to such programs. *Id.* at 232a-233a. The district court also concluded that the 1-2-3 menus, considering their contents, arrangement, and presentation on the screen, contained expression that was not essential to the "idea" of a spreadsheet menu structure, and were only one of "many if not an unlimited" number of ways to express that idea. *Id.* at 234a. Finding that Paperback had copied these protected elements and that these elements were qualitatively substantial (*id.* at 238a-239a), the district court concluded that Paperback had infringed.

In *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), the Second Circuit strongly criticized the Third Circuit's approach in *Whelan* for seeming to assume that "only one 'idea' . . . underlies any computer program, and that once a separable idea can be identified, everything else must be expression.'" *Id.* at 705, *quoting from* David Nimmer & Melville B. Nimmer, 3 *Nimmer on Copyright*, § 13.03[F][1] at 13-62.34

(hereinafter "Nimmer"). The *Altai* court adopted the "abstraction-filtration-comparison" test proposed in Nimmer, the leading treatise in this field, for determining whether an infringement of the non-literal elements of computer programs has occurred. Like the *Paperback* test, the *Altai* test also proceeds in three steps, the first of which is an "abstractions" analysis derived from Judge Hand's *Nichols* decision. 982 F.2d at 706-07. In the second step of "filtration," the court examines each element of the copyrighted work to "screen" out those that are functionally dictated, are standard to the treatment of the subject, or are not original. *Id.* at 707. This step serves to define the scope of plaintiff's copyright. *Id.* In the final step of "comparison," the court compares the elements of the copyrighted work that survive the filtration process to corresponding elements of the allegedly infringing work to determine whether enough copying has occurred to find infringement. *Id.* at 710-11.[13] The *Altai* court cited *Paperback* with approval in its analysis. *Id.* at 702. According to the Nimmer treatise, the *Altai* and *Paperback* tests, even though phrased differently, "harmonize" and yield similar results. 3 Nimmer, § 13.03[F][1] at 13-131, n.303.13.

In *Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, 9 F.3d 823, 840-41 (10th Cir. 1993), the Tenth Circuit relied upon the *Whelan*, *Paperback*, and *Altai* analyses to formulate its own variation of the "abstraction-filtration-comparison" test. The court noted the *Altai* court's criticism of the *Whelan* decision, but stated that "when a program is understood to encompass more than one idea, the general principle of *Whelan* provides a useful means to distinguish idea from expression," and found "its conclusion that the structure of a program may be protectable is sound." *Id.* at 840. Referring to Judge Keeton as having "written extensively on the subject of software protection," the court described the *Paperback* test as a "forerunner of the standard that we adopt in this case." *Id.* at 840-41.

The *Paperback* test also was cited with approval and relied upon by the Fifth Circuit in *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1343 (5th Cir. 1994). The Fifth Circuit expressly endorsed the *Gates Rubber* "abstraction-filtration-comparison method for determining copyright protection for computer programs." *Id.* at 1342. The court described Judge Keeton's test as a "similar systematic approach." *Id.* at 1343. Thus, prior to the First Circuit's judgment in this case, the courts of appeals appeared to be converging on an appropriate methodology for applying the idea/expression distinction to non-literal elements of computer programs, based in large part upon Judge Keeton's decisions.

[13] Even though Judge Keeton was not required to perform a "comparison" step in this or the *Paperback* action, because in both cases the defendants admitted to copying the elements at issue, he did determine that those elements were a qualitatively substantial part of 1-2-3.

C. Prior Proceedings in This Case

1. In the District Court

The district court issued the *Paperback* opinion on June 28, 1990. Lotus filed this action four days later. Both parties moved for summary judgment. The district court granted Lotus' motion in part and denied Borland's motion.

The district court concluded that Borland essentially had admitted to having "copied the menu commands and command structure of Lotus 1-2-3." Pet. App. at 113a. Applying the *Paperback* test as modified in an earlier decision in this case (*id.* at 163a-165a), Judge Keeton found that the appropriate conception of the idea behind the 1-2-3 menu command hierarchy was a "system of menus," hierarchically arranged as in 1-2-3, "so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through" the menu command hierarchy. *Id.* at 129a. The district court found that "literally millions of satisfactory menu trees" could be generated to express this idea (*id.* at 131a), and concluded that it was beyond genuine dispute "that a large part of the structure and arrangement of the [1-2-3] menu commands is not driven entirely by functional considerations." *Id.* at 133a. The district court also determined, however, that Borland had identified potential issues for trial concerning whether some aspects of the 1-2-3 menus were "functionally dictated" by certain "functional rules" or by concerns of "efficiency and usefulness." *Id.* at 116a.

Following the summary judgment decision, Borland removed the "1-2-3 emulation" menu trees from its products and began to publicize the existence of the "Key Reader." *Id.* at 54a-56a. Lotus sought and was granted leave to file a supplemental complaint alleging that the Key Reader also infringed its 1-2-3 copyrights. *Id.* The district court held two bench trials after Borland waived its jury demand. *Id.* at 75a-77a. The scope of the Phase I trial was defined by stipulation as "all issues not previously *finally* decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto," excluding Key Reader issues. *Id.* at 75a. The Phase II trial addressed all issues relating to the Key Reader. *Id.* at 76a.

The district court issued separate opinions concerning the two phases of trial. In the Phase I opinion, the court held that "what Borland copied from 1-2-3 was not limited to aspects dictated by functional constraints. Rather, Borland copied the entire menu tree, much of which was the free expression of the creators of Lotus 1-2-3." *Id.* at 89a. In addition, the district court examined the 1-2-3 menu command hierarchy pursuant to the "conceptual separability" test articulated by this Court in *Mazer v. Stein*, 347 U.S. 201, 218 (1954), and found as fact that the 1-2-3 menu command hierarchy contained expressive elements that were separable from its functional aspects or ultimate uses, and was therefore copyrightable.[14] *Id.* at 93a & 130a. The Phase II opinion held that the Key Reader's "phantom" menus contained a "virtually identical" copy of "details of expression of the Lotus 1-2-3 program's menu structure." *Id.* at 46a-47a. Concluding that Borland had infringed Lotus' copyrights, the district court entered a permanent injunction (at Borland's request) prohibiting further sales of Borland's products in a form that contained the infringing features. *Id.* at 69a-70a.

[14] Even though the district court acknowledged that, as codified in the 1976 Copyright Act, the "conceptual separability" test applies to "useful articles" such as lamp bases falling within the definition of "pictorial, graphical, and sculptural works," rather than to "literary works" such as computer programs, Judge Keeton nonetheless applied this stricter test to the Lotus 1-2-3 menu command hierarchy. Pet. App. at 117a. *See* 17 U.S.C. § 101 (definitions of "useful article" and of "pictorial, graphical, and sculptural works").

2. In the Court of Appeals

The First Circuit approached the question of interpretation of Sections 102(a) and 102(b) in a fundamentally different way than did the district court. Describing the case as one of "first impression in this court" (Pet. App. at 12a), the majority acknowledged that its approach conflicted with that followed by other courts of appeals. *Id.* at 21a-22a. To the First Circuit, the "initial inquiry" was *not* "whether the Lotus menu command hierarchy incorporates expression." *Id.* at 17a. Indeed, the majority accepted the district court's finding that it did. *Id.* The majority also considered the potentially very large number of ways to express that menu command hierarchy to be "immaterial" to its analysis, again accepting the fact that numerous alternative expressions did exist. *Id.*

The majority instead read Section 102(b) as enumerating a set of "categories" of works "foreclosed from copyright protection," even if they contain original expression that would otherwise qualify for copyright protection under Section 102(a). *Id.* at 21a. The "initial inquiry" for a court, according to the First Circuit, is to determine whether a work can be characterized as fitting into one of these categories, such as "method of operation." *Id.* at 17a. The court below opined that the 1-2-3 menu command hierarchy is a "method of operation" because it "provides the means by which users control and operate Lotus 1-2-3." *Id.* at 15a.[15] Concluding that "the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3" (*id.* at 17a-18a) by analogy to the buttons on a videocassette recorder ("VCR") machine (*id.* at 18a-19a), the court of appeals saw no need to inquire further into whether or to what extent the menu command hierarchy reflected " 'expressive' choices" or was capable of alternative expression. *Id.* at 17a.[16] The First Cir-

[15] Judge Boudin's concurrence describes this "formulation" as "defensible" upon the authority of *Webster's College Dictionary*. *Id.* at 27a. The majority cites to no authority for its interpretation.

[16] The First Circuit did not mention, or apparently consider, the district court's finding that the 1-2-3 menu command hierarchy contained copyrightable expression even under the "conceptual separability" test applicable to "useful articles" such as lamp bases, as articulated by this

cuit found its conclusion to be "bolstered" by this Court's ruling in *Baker v. Selden*, 101 U.S. 99 (1879). Pet. App. at 18a. Because Lotus wrote the menu command hierarchy "so that people could learn it and use it," the court below declared that it "falls squarely within the prohibition on copyright protection established" in that case and codified in Section 102(b). *Id.*

Judge Boudin's concurrence expresses the view that Congress's role in the copyright context is limited to sketching the "broad-brush conception" and prescribing "formalities." *Id.* at 25a. The "heart of copyright doctrine—what may be protected and with what limitations and exceptions"—is, according to him, left by "tradition" to be determined by the courts. *Id.* Judge Boudin recognized that the majority's reading of Section 102(b) could "exclude most computer programs from protection" and that the statute could be read (and has been by other courts) "in cookie cutter fashion" as a "congressional command" to the contrary. *Id.* at 24a-25a. Nevertheless, he agreed with the majority's result for reasons more appropriate to antitrust analysis than to copyright—in particular his concern about users being "locked into Lotus" as a "*de facto* standard for electronic spreadsheet commands." *Id.* at 26a.[17] Finding the majority's "formulation is as good, if not better, than any other that occurs to me now as within the reach of courts" (*id.* at 28a), he declared that "the choices are important ones of policy, not linguistics, and they should be made with the underlying considerations in view." *Id.* In Judge Boudin's view, Congress did not resolve these important policy choices, but instead left the courts a free hand to do what they thought best.[18]

Court in *Mazer v. Stein*, 347 U.S. 201, 218 (1954), and codified in the 1976 Copyright Act.

[17] Judge Boudin cites nothing in the record below to support his factual premises and assumptions concerning Lotus' market position and the competitive effect of enforcing Lotus' copyrights (issues that were *not* raised or tried in the district court), Borland's motives for copying, or the extent to which users' purchases of Quattro Pro proved it to be a "better product," rather than just a cheaper imitation and substitute. *Id.* at 26a-27a.

[18] Judge Keeton expressed a very different view concerning the extent to which judicial action in this field is constrained by Congressional mandate. *Id.* at 166a & 206a-207a. *Accord, Altai*, 982 F.2d at 702 (the "statutory terrain in this area has been well explored," citing *Paper-*

REASONS FOR GRANTING THE PETITION

This Court has stated that the copyright law, following explicit Constitutional authority, reflects a belief that the limited monopolies it confers serve the public welfare by encouraging authors to generate new ideas and to disseminate them to the public in any originally expressed way that they may choose. *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 546 (1985). Although the "immediate effect of our copyright law is to secure a fair return for an 'author's' creative labor," the ultimate aim of this incentive is to stimulate creativity for the "general public good." *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 432 (1984), *quoting from Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975). The policy of "encouragement of individual effort by personal gain" has long been viewed as "the best way to advance public welfare through talents of authors and inventors in 'Science and useful Arts'." *Mazer v. Stein*, 347 U.S. 201, 219 (1954). When Congress decided in 1976 to treat computer programs as copyrightable literary works, and reaffirmed that decision by embracing the CONTU recommendations in 1980, it manifested a policy judgment to promote progress in this nascent field of intellectual creativity by extending to the authors of these new forms of writing the same encouragement that all other authors receive. Whatever form of substantive regime this policy may dictate, it is essential to its realization that it be applied in a uniform manner.

Until the First Circuit's decision in this case, the various courts of appeals had reached an apparent consensus on certain fundamental principles regarding copyright protection for computer software. *First*, that Congress intended that copyright protection apply to computer programs under the same principles governing other "literary works." *Second*, that because Section 102(b) is a codification of the idea/expression distinction, any attempt to delineate the extent to which copyright will protect computer programs must be grounded in that distinction. *Third*, that the courts should extend the scope of protection for computer programs, as

back, and the legislative history "leaves no doubt that Congress intended" computer programs to be treated like other literary works).

with other types of literary works, beyond their literal manifestations to their non-literal elements as well, to the extent they reflect the author's original expression.

In the shaping of this consensus, the five opinions of Judge Keeton in this and the *Paperback* case have become a touchstone—a common citation and point of departure in any analysis of this issue. His opinions are cited with approval by the Second, Ninth, and Tenth Circuits, and relied upon explicitly by the Fifth.[19] Not one of these courts has suggested that the central tenets of Judge Keeton's analysis have been anything but correct. If there is divergence, it is only on points of emphasis or in the details of the particular methodology to follow in applying that analysis to the facts of an individual case.

The First Circuit's relatively brief opinion comes as a jarring departure from the prevailing consensus and brushes all this painstaking work aside. The First Circuit quarrels with everybody: with the district court, with the Nimmer treatise, with Learned Hand, and with the Second, Ninth, and Tenth Circuits. Pet. App. at 14a-15a. And because the Fifth Circuit in *Engineering Dynamics* explicitly adopts Judge Keeton's analysis, the First Circuit must disagree with that circuit as well. The result is conflict both in outcome and approach, as the First Circuit openly acknowledges. *Id.* at 21a-22a. This alone justifies this Court's review.

But the potential consequences of the First Circuit's decision run deeper still. Under the First Circuit's reasoning, it is unclear what, if any, elements of computer programs would merit protection, because all programs to some degree describe a "method of operation" for a machine—as Judge Boudin concedes. This presents a serious problem for an important American industry. In the years since the adoption of the Copyright Act in 1976 and the software amendments of 1980, the computer software industry has prospered, relying primarily, as Congress no doubt expected, upon copyright law to protect the fruits of its creative efforts. Although the analogy between novels or plays and these

[19] Judge Keeton's opinions below have been favorably cited in copyright cases by more than 20 different Federal trial and appellate courts.

new digital works of authorship occasionally may seem strained (no more so, of course, than with lamp bases), the growth of the industry, both in size and diversity, strongly suggests that Congress's policy decision to protect computer programs under the copyright law has worked well. The First Circuit's decision potentially destabilizes a significant portion of accepted copyright doctrine as it has been applied by the courts to computer programs, following the Congressional mandate.

I

THE DECISION OF THE COURT OF APPEALS CONFLICTS WITH DECISIONS OF NO FEWER THAN FIVE OTHER CIRCUITS

The First Circuit's rejection of the idea-expression distinction in interpreting Section 102(b) and its holding that menu command hierarchies are an uncopyrightable "method of operation" conflicts with decisions of no fewer than five other circuits. Most recently, in *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1343 (5th Cir. 1994), the Fifth Circuit held that plaintiff's selection of "approximately 230 input-output formats that comprise the user interface" of a structural engineering program, "taken as a whole, readily qualify as 'expression' measured against the ideas versus expression dichotomy." *Id.* at 1343-44. These formats serve "to mediate between the user and the program, identifying what information is essential and how it must be ordered to make the program work." *Id.* at 1344. Under the First Circuit's reasoning, these characteristics would render the formats part of an unprotected "method of operation." To the Fifth Circuit, they served as indicia that the formats conveyed more than an unprotectable "idea."

The *Engineering Dynamics* court relied heavily upon Judge Keeton's opinions in *Paperback* and this case. Upon the authority of *Paperback*, it concluded that "the command format and sequence structure in an original word processing or computer spreadsheet should be copyrightable" because, as a whole, they contain "a high degree of original expression." *Id.* at 1345-46. The court recognized that the input formats at issue "ultimately act like switches in the electrical circuits of the program," but

concluded that this utilitarian function did not "outweigh their expressive purpose so as to preclude copyright protection." *Id.* at 1346. The court went so far as to express incredulity that the law would permit blatant copying of a best-selling program's user interface (*id.*)—the very conduct that the First Circuit not only condoned in this case, but celebrated as socially desirable. Pet. App. at 20a-21a & 26a-27a.

The First Circuit's decision also conflicts with the Tenth Circuit's ruling in *Autoskill, Inc. v. National Educational Support Systems, Inc.*, 994 F.2d 1476 (10th Cir.), *cert. denied*, ___ U.S. ___, 114 S. Ct. 307 (1993), as the First Circuit explicitly acknowledged. Pet. App. at 21a. In *Autoskill*, the plaintiff's program comprised a "system" (so described) for testing reading skills in a structured, query-and-response format. 994 F.2d at 1481. The queries that the program displayed to the user consisted of a selection of individual words, arranged in sequence according to various skill types and levels, that the Tenth Circuit held to be copyrightable. *Id.* at 1495-96. The Tenth Circuit also found that the program's "keying procedure" by which users communicated responses to the program was sufficiently expressive, applying the merger doctrine and the "idea/expression" distinction, to avoid classification as a "method of operation" under Section 102(b). *Id.* The First Circuit declares that it would have ruled otherwise. Pet. App. at 22a. The conflict is plain.[20]

The First Circuit concedes that its decision also can be read to conflict with rulings of the Ninth Circuit (Pet. App. at 22a), which has indicated on two occasions that it considers "the non-literal components of a program," including the "manner in which information is presented to the user" in a user interface, to constitute copyrightable subject matter upon a showing that

[20] The First Circuit further conflicts with the Tenth Circuit concerning the proper approach to interpreting Section 102(b). In *Autoskill*, that court declared that "[w]e must go beyond the literal language of the statute and apply the idea/expression distinction to resolve" the copyrightability issue. 994 F.2d at 1495 n.23. *See also Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, 9 F.3d 823, 849 (10th Cir. 1993) (directing trial court to determine the copyrightability of "menus and sorting criteria" in plaintiff's program under a variation of the "abstraction-filtration-comparison" test rejected by the First Circuit).

the "component in question qualifies as an expression of an idea." *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 n.3 (9th Cir. 1989). *Accord, Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1476-77 (9th Cir.), *cert. denied sub nom. BB Asset Mgmt. Inc. v. Symantec Corp.*, ___ U.S. ___, 113 S. Ct. 198 (1992) (recognizing protectability of screens, menus, and keystrokes of the program's user interface). As the First Circuit observed (Pet. App. at 22a), the holding in *Brown Bag*, narrowly construed, was to affirm a finding of no infringement because the defendant had not copied the plaintiff's menus or keystrokes. 960 F.2d at 1475. Nonetheless, a conflict exists because the premise of the Ninth Circuit's holding is that the menus and keystrokes were protected and therefore susceptible of being infringed, had they been copied. Moreover, the Ninth Circuit has declared that its test for determining the copyrightability of non-literal elements of computer programs, "although articulated differently," is similar to that applied by the district court and overturned by the First Circuit in this case. *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1445-46 (9th Cir. 1994), *cert. denied*, ___ U.S. ___, 115 S. Ct. 1176 (1995).

The First Circuit's interpretation of Section 102(b) as applied to computer programs also conflicts with that of the Second and Third Circuits. The court below specifically rejected the "abstraction-filtration-comparison" test adopted by the Second Circuit in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 703-10 (2d Cir. 1992), because it perceived that test as resting upon an assumption that at least *some* element of the work being examined is potentially protectable. Pet. App. at 15a.[21] Indeed, the First Circuit rejects not only the *Altai* test but also Judge Hand's "abstractions" analysis from *Nichols*, declaring that the abstraction process "obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all." *Id.*

[21] The First Circuit faults the "abstraction-filtration-comparison" test generally "because it seems to encourage [courts] to find a base level [of abstraction] that includes copyrightable subject matter," and implies that every case of literal similarity would result in a finding of infringement. *Id.* at 14a.

Finally, although the First Circuit does not mention it, the decision in *Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987), remains good law in the Third Circuit, despite any criticism it has received elsewhere. Because the First Circuit finds the *Altai* test too lenient, and the *Altai* test was in turn a response to perceived limitations of *Whelan* (982 F.2d at 705), it must follow that the First Circuit also rejects the *Whelan* approach. More fundamentally, the interpretation given to Section 102(b) by the court below conflicts with that established in the Third Circuit over ten years ago in *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d at 1240, 1250-52 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984). In that case, the Third Circuit rejected an argument that an operating "system" was "*per se* exclud[ed] from copyright protection under the express terms" of Section 102(b) (*id.* at 1250), finding instead that the "expression/idea dichotomy is now expressly recognized" in that provision. *Id.* at 1252. And unlike the Third Circuit, which refused to accept a purported "compatibility" defense it described as a "commercial and competitive objective" that did not enter into its copyrightability determination (*id.* at 1253), the First Circuit expressly relied upon Borland's professed goal of achieving "program compatibility" to support its conclusion that the 1-2-3 menu command hierarchy was an uncopyrightable "method of operation." Pet. App. at 20a.

In sum, the court below deliberately has set itself in conflict, both in outcome and approach, with the decisions of no fewer than five other circuits. As revealed by Judge Boudin's comment that "no intermediate appellate court can make the final choice" as to the "right" solution (*id.* at 27a), the First Circuit appears to have gone out of its way to invite this Court's review.

II

THE QUESTION OF FEDERAL LAW DECIDED BY THE COURT OF APPEALS UNSETTLES THE LAW IN AN IMPORTANT AREA AND THEREFORE THE PUBLIC INTEREST REQUIRES THIS COURT'S REVIEW

Both the statute and the legislative history manifest Congress's intent to protect computer programs as literary

works under the copyright law. Congress considered but rejected the establishment of special rules to apply to these new digital works of authorship.[22] In so doing, Congress was fully aware of the functional nature of computer programs. The definition it adopted for them in the Copyright Act—"a set of statements or instructions to be *used* directly or indirectly in a computer *in order to bring about a certain result*" (17 U.S.C. § 101; emphasis supplied)—is ample proof, standing alone, that Congress understood that computer programs were expressive works much of whose value would lie in their utility rather than their aesthetic appeal. Congress placed its faith in the courts to give meaningful protection to programs, despite whatever difficulties and doctrinal rough edges might arise from its decision to recognize these admittedly functional works within the body of copyrightable subject matter. House Report at 57, *reprinted in* 1976 U.S. Code Cong. & Admin. News at 5670. Its decision is reflected in its specific reference to Section 102(b), which it explained was meant to restate existing case law concerning the dichotomy between expression and idea, no more and no less ("Section 102(b) . . . in no way enlarges or contracts the scope of copyright protection under present law."). *Id.*

The First Circuit's interpretation of Section 102(b) as establishing a "string of exclusions" (Pet. App. at 24a), definitional in nature, through which an element of a work must pass *before* the expressive aspects of that element may be protected, is at odds with everything that has preceded it. Since Congress acted, certain basic parameters and principles concerning the scope of protection for computer programs have been established by the Federal appellate and trial courts as they have worked over time, in response to the distinct facts of individual cases, to fulfill the Congressional mandate. None has interpreted Section 102(b) in this manner.

In the wake of the First Circuit's decision, software developers (and their investors) can no longer tell whether, or to what extent, their creative efforts will receive effective protection or

---

[22] The only unique limitation imposed on the rights conferred on authors of computer programs was the grant of limited rights to program users to modify or make back-up copies (for personal use only) of other legitimately-acquired copies. 17 U.S.C. § 117.

may encroach upon the rights of others. In the software industry, product design decisions often are part of multi-million dollar research and development programs, followed by equally large marketing expenditures, all intended to generate sales across a national (and international) market. Uncertainty as to precisely where the line is drawn may be a long-standing problem in copyright law,[23] but when fortuities of forum and venue can lead to flatly inconsistent outcomes and analyses, an industry operating on a nationwide basis simply cannot order its affairs in accordance with the rule of law.

One obvious and important area of legal uncertainty is the extent to which a program's menus and other original, expressive elements of its user interface are eligible for copyright protection. As the software industry has flourished and the programming art has advanced, the creative efforts of software developers have become increasingly focused upon the design and implementation of user interfaces that make it easier for users to understand and put to productive use the power of computers. But the ability of developers to protect their creations from blatant copying is now in doubt. It is now legal to copy menus in the First Circuit, but not in the Fifth, Ninth, or Tenth. Without the uniformity of law that only this Court can achieve, the industry and users are left to speculate as to whether a particular product or application will receive protection or whether another will be deemed an infringement. Rational product development and investment decisions, and the continued growth of this vital industry, depend upon this Court providing a clear and final resolution of this issue.

The uncertainty created by the First Circuit's decision extends beyond the copyrightability of menu command hierarchies and other expressive elements of user interfaces. If the characterization of a work as a "method of operation," despite the presence of separable expression, is sufficient to defeat its copyrightability, then all elements of computer programs (except, perhaps, their decorative or ornamental features)—even

---

[23] *See Peter Pan Fabrics v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960) (L. Hand, J.) (decisions as to when an imitator has gone beyond the idea and has borrowed its expression must inevitably be *ad hoc*); *Paperback*, Pet. App. at 244a-246a.

their source or object code—stand on shaky ground. As is true of menu command hierarchies, a program consists of a set of statements and instructions that achieve a functional result when communicated to a computer. Indeed, the very purpose of a computer program is to express a "method" or "process" or "system" by which a user can operate a computer to perform some functional task; the same may be said of any manual or instructional text. Whether or not a particular expression of that "method" is "essential" to use of the "method" becomes, under the First Circuit's reasoning, largely irrelevant: it depends merely on how the "method" is defined. The First Circuit's decision provides no guidance for making this determination beyond its analogies to (uncopyrightable) machines such as VCR's and food processors.

It is difficult to discern a logical boundary in the First Circuit's approach to limit the ways in which its ruling, if followed, could serve to roll back the scope of protection for computer programs generally. Carrying the First Circuit's reasoning slightly further towards its logical conclusion, for example, even the Third Circuit's seminal decision in *Apple v. Franklin*, which first extended protection to operating system software, would be called into question. Such software certainly provides a "means by which users control and operate" a computer, and thus comprises a "method of operation." Yet, without the expression in the operating system's set of instructions, no use can be made of its functional capabilities, or of programs designed to work with that operating system. It is not at all clear under what rationale the First Circuit could now sustain the *Apple v. Franklin* holding.

Thus, Judge Boudin is correct in observing that the issues raised in this case are "important ones of policy." Pet. App. at 28a. Whether the First Circuit's novel interpretation of Section 102(b) is consistent with the statement and intentions of Congress; whether copying of user interfaces serves the public interest; whether copyright law and policy as articulated by this Court over the years should be limited, when applied to computer programs, by competition policy considerations and, if so, upon what basis, and by Congress or by the courts; and whether copyright in computer programs should be construed more nar-

rowly, more critically or simply differently than in other forms of literary works, are all questions raised by the First Circuit's decision that bear upon the extent to which computer programs will receive meaningful and effective copyright protection in the future.

Finally, the First Circuit's unique view that Section 102(b) somehow "trumps" the statutory protection accorded the expressive elements of copyrightable works by Section 102(a) is a conception that is not limited to computer programs, or even literary works. It would reach all copyrightable subject matter and has the potential to undo a generation of copyright precedent. *Cf. Mazer v. Stein*, 347 U.S. 201, 218 (1954). This Court's guidance, therefore, seems imperative.

Conclusion

The petition for a writ of certiorari should be granted.

Respectfully submitted,

CHARLES FRIED
1545 Massachusetts Avenue
Cambridge, MA 02138
(617) 495-4111

ARTHUR R. MILLER
1575 Massachusetts Avenue
Cambridge, MA 02138
(617) 495-4636

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, MA 02142
(617) 693-8106

Counsel for Petitioner

June 7, 1995

Supreme Court, U.S.
FILED

No. _____

942003 JUN 7 1995

OFFICE OF THE CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

APPENDIX TO PETITION FOR A WRIT OF CERTIORARI

CHARLES FRIED
1575 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4636

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

Counsel for Petitioner

INDEX TO APPENDIX

PAGE

Opinion of the United States Court of Appeals for the First Circuit in *Lotus Development Corporation v. Borland International, Inc.*, dated March 9, 1995. 1a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 12, 1993, as amended August 19, 1993 ... 29a

Permanent Injunction entered by United States District Court of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 19, 1993 ..... 69a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated June 30, 1993, as amended August 19, 1993 ..... 71a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated July 31, 1992 ..... 106a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated March 20, 1992..... 145a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Paperback Software International*, dated June 28, 1990 ..... 183a

PAGE

The 1976 Copyright Act, 17 U.S.C. §§ 101, 102(a), 102(b), 103, 117 ..... 270a

Affidavit of Mitchell D. Kapor, dated November 27, 1989, submitted in *Lotus Development Corporation v. Borland International, Inc.* and *Lotus Development Corporation v. Paperback Software International* 281a

UNITED STATES COURT OF APPEALS
FOR THE FIRST CIRCUIT

No. 93-2214

LOTUS DEVELOPMENT CORPORATION,

Plaintiff, Appellee,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant, Appellant.

Appeal from the United States District Court
for the District of Massachusetts
[Hon. Robert E. Keeton *U.S. District Judge*]

Before
TORRUELLA, *Chief Judge*,
BOUDIN and STAHL, *Circuit Judges*.

Gary L. Reback, with whom *Peter N. Detkin, Michael Barclay, Isabella E. Fu, Wilson Sonsini Goodrich & Rosati, Peter E. Gelhaar, Katherine L. Parks*, and *Donnelly Conroy & Gelhaar*, were on brief for appellant.

Matthew P. Poppel, et al., were on brief for Computer Scientists, amicus curiae.

Dennis S. Karjala, and *Peter S. Menell*, on brief, amici curiae.

Jeffrey C. Cannon and *Baker Keaton Seibel & Cannon*, were on brief for Computer Software Industry Ass'n, amicus curiae.

Laureen E. McGurk, David A. Rabin, Bryan G. Harrison and *Morris Manning & Martin*, were on brief for Chicago Computer Soc., Diablo Users Group, Danbury Area Computer Soc., IBM AB Users Group, Kentucky-Indiana Personal Computer Users Group, Long Island PC Users Group, Napa Valley PC Users Group, Pacific Northwest PC Users Group, Palmetto Personal Computer Club, Philadelphia Area Computer Soc., Inc., Phoenix IBM PC Users Group, Pinellas IBM PC Users Group, Quad Cities Computer Soc., Quattro Pro Users Group, Sacramento PC Users Group, San Francisco PC Users Group, Santa Barbara PC Users Group, Twin Cities PC Users Group, and Warner Robbins Personal Computer Ass'n, amici curiae.

Diane Marie O'Malley and *Hanson Bridgett Marcus Vlahos & Rudy*, were on brief for Software Entrepreneurs' Forum, amicus curiae.

Peter M.C. Choy, was on brief for American Committee for Interoperable Systems, amicus curiae.

Howard B. Abrams, Howard C. Anawalt, Stephen R. Barnett, Ralph S. Brown, Stephen L. Carter, Amy B. Cohen, Paul J. Heald, Peter A. Jaszi, John A. Kidwell, Edmund W. Kitch, Roberta R. Kwall, David L. Lange, Marshall Leaffer, Jessica D. Litman, Charles R. McManis, L. Ray Patterson, Jerome H. Reichman, David A. Rice, Pamela Samuelson, David J. Seipp, David E. Shipley, Lionel S. Sobel, Alfred C. Yen, and Diane L. Zimmerman, were on brief for Copyright Law Professors, amicus curiae.

Henry B. Gutman, with whom *Kerry L. Konrad, Joshua H. Epstein, Kimberly A. Caldwell, O'Sullivan Graev & Karabell, Thomas M. Lemberg, James C. Burling*, and *Hale and Dorr*, were on brief for appellee.

Morton David Goldberg, June M. Besek, David O. Carson, Jesse M. Feder, Schwab Goldberg Price & Dannay, and

Arthur R. Miller, were on brief for Apple Computer, Inc., Digital Equip. Corp., International Business Machines Corp., and Xerox Corp., amici curiae.

Jon A. Baumgarten, Proskauer Rose Goetz & Mendelsohn, and *Robert A. Gorman*, were on brief for Adobe Systems, Inc., Apple Computer, Inc., Computer Associates Intern., Inc., Digital Equip. Corp., and International Business Machines Corp., amici curiae.

Herbert F. Schwartz, Vincent N. Palladino, Susan Progoff, Fish & Neave, William J. Cheeseman, and *Foley Hoag & Eliot*, were on brief for Computer and Business Equip. Mfrs. Ass'n, amicus curiae.

STAHL, *Circuit Judge*.

This appeal requires us to decide whether a computer menu command hierarchy is copyrightable subject matter. In particular, we must decide whether, as the district court held, plaintiff-appellee Lotus Development Corporation's copyright in Lotus 1-2-3, a computer spreadsheet program, was infringed by defendant-appellant Borland International, Inc., when Borland copied the Lotus 1-2-3 menu command hierarchy into its Quattro and Quattro Pro computer spreadsheet programs. *See Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass. 1992) ("*Borland I*"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F.Supp. 203 (D.Mass. 1992) ("*Borland II*"); *Lotus Dev. Corp. v. Borland Int'l; Inc.*, 831 F.Supp. 202 (D.Mass. 1993) ("*Borland III*"); *Lotus Dev. Corp. v. Borland Int'l Inc.*, 831 F.Supp. 223 (D.Mass. 1993) ("*Borland IV*").

I.

Background

Lotus 1-2-3 is a spreadsheet program that enables users to perform accounting functions electronically on a computer. Users manipulate and control the program via a series of menu commands, such as "Copy," "Print," and "Quit." Users choose commands either by highlighting them on the screen or by typing their first letter. In all, Lotus 1-2-3 has 469 commands arranged into more than 50 menus and submenus.

Lotus 1-2-3, like many computer programs, allows users to write what are called "macros." By writing a macro, a user can designate a series of command choices with a single macro keystroke. Then, to execute that series of commands in multiple parts of the spreadsheet, rather than typing the whole series each time, the user only needs to type the single preprogrammed macro keystroke, causing the program to recall and perform the designated series of commands automatically. Thus, Lotus 1-2-3 macros shorten the time needed to set up and operate the program.

Borland released its first Quattro program to the public in 1987, after Borland's engineers had labored over its development for nearly three-years. Borland's objective was to develop a spreadsheet program far superior to existing programs, including Lotus 1-2-3. In Borland's words, "[f]rom the time of its initial release . . . Quattro included enormous innovations over competing spreadsheet products."

The district court found, and Borland does not now contest, that Borland included in its Quattro and Quattro Pro version 1.0 programs "a *virtually identical* copy of the entire 1-2-3 menu tree." *Borland III*, 831 F.Supp. at 212 (emphasis in original). In so doing, Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy. Borland included the Lotus menu command hierarchy in its programs to make them compatible with Lotus 1-2-3 so that spreadsheet users who were already familiar with Lotus 1-2-3 would be able to

switch to the Borland programs without having to learn new commands or rewrite their Lotus macros.

In its Quattro and Quattro Pro version 1.0 programs, Borland achieved compatibility with Lotus 1-2-3 by offering its users an alternate user interface, the "Lotus Emulation Interface." By activating the Emulation Interface, Borland users would see the Lotus menu commands on their screens and could interact with Quattro or Quattro Pro as if using Lotus 1-2-3, albeit with a slightly different looking screen and with many Borland options not available on Lotus 1-2-3. In effect, Borland allowed users to choose how they wanted to communicate with Borland's spreadsheet programs: either by using menu commands designed by Borland, or by using the commands and command structure used in Lotus 1-2-3 augmented by Borland-added commands.

Lotus filed this action against Borland in the District of Massachusetts on July 2, 1990, four days after a district court held that the Lotus 1-2-3 "menu structure, taken as a whole—including the choice of command terms [and] the structure and order of those terms," was protected expression covered by Lotus's copyrights. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 68, 70 (D.Mass.1990) ("*Paperback*").[1] Three days earlier, on the morning after the *Paperback* decision, Borland had filed a declaratory judgment action against Lotus in the Northern District of California, seeking a declaration of non-infringement. On September 10, 1990, the district court in California dismissed Borland's declaratory judgment action in favor of this action.

Lotus and Borland filed cross motions for summary judgment; the district court denied both motions on March 20, 1992, concluding that "neither party's motion is supported by the record." *Borland I*, 788 F.Supp. at 80. The district court invited the parties to file renewed summary judgment motions that would "focus their arguments more precisely" in light of rulings it had made in conjunction with its denial of their summary judgment motions. *Id.* at 82. Both parties filed

---

[1] Judge Keeton presided over both the *Paperback* litigation and this case.

renewed motions for summary judgment on April 24, 1992. In its motion, Borland contended that the Lotus 1-2-3 menus were not copyrightable as a matter of law and that no reasonable trier of fact could find that the similarity between its products and Lotus 1-2-3 was sufficient to sustain a determination of infringement. Lotus contended in its motion that Borland had copied Lotus 1-2-3's entire user interface and had thereby infringed Lotus's copyrights.

On July 31, 1992, the district court denied Borland's motion and granted Lotus's motion in part. The district court ruled that the Lotus menu command hierarchy was copyrightable expression because

> [a] very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

Borland II, 799 F.Supp. at 217. The district court demonstrated this by offering alternate command words for the ten commands that appear in Lotus's main menu. *Id.* For example, the district court stated that "[t]he 'Quit' command could be named 'Exit' without any other modifications," and that "[t]he 'Copy' command could be called 'Clone,' 'Ditto,' 'Duplicate,' 'Imitate,' 'Mimic,' 'Replicate,' and 'Reproduce,' among others." *Id.* Because so many variations were possible, the district court concluded that the Lotus developers' choice and arrangement of command terms, reflected in the Lotus menu command hierarchy, constituted copyrightable expression.

In granting partial summary judgment to Lotus, the district court held that Borland had infringed Lotus's copyright in Lotus 1-2-3:

> [A]s a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

Borland II, 799 F.Supp. at 223 (emphasis in original). The court nevertheless concluded that while the Quattro and Quattro Pro programs infringed Lotus's copyright, Borland had not copied the entire Lotus 1-2-3 user interface, as Lotus had contended. Accordingly, the court concluded that a jury trial was necessary to determine the scope of Borland's infringement, including whether Borland copied the long prompts[2] of Lotus 1-2-3, whether the long prompts contained expressive elements, and to what extent, in any, functional constraints limited the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See Borland III*, 831 F.Supp. at 207. Additionally, the district court granted Lotus summary judgment on Borland's affirmative defense of waiver, but not on its affirmative defenses of laches and estoppel. *Borland II*, 799 F.Supp. at 222-23.

---

[2] Lotus 1-2-3 utilizes a two-line menu; the top line lists the commands from which the user may choose, and the bottom line displays what Lotus calls its "long prompts." The long prompts explain, as a sort of "help text," what the highlighted menu command will do if entered. For example, the long prompt for the "Worksheet" command displays the submenu that the "Worksheet" command calls up; it reads "Global, Insert, Delete, Column, Erase, Titles, Window, Status, Page." The long prompt for the "Copy" command explains what function the "Copy" command will perform: "Copy a cell or range of cells." The long prompt for the "Quit" command reads, "End 1-2-3 session (Have you saved your work?)."

Prior to trial, the parties agreed to exclude the copying of the long prompts from the case; Lotus agreed not to contend that Borland had copied the long prompts, Borland agreed not to argue that it had not copied the long prompts, and both sides agreed not to argue that the issue of whether Borland had copied the long prompts was material to any other issue in the case. *See Borland III*, 831 F.Supp. at 208.

Immediately following the district court's summary judgment decision, Borland removed the Lotus Emulation Interface from its products. Thereafter, Borland's spreadsheet programs no longer displayed the Lotus 1-2-3 menus to Borland users, and as a result Borland users could no longer communicate with Borland's programs as if they were using a more sophisticated version of Lotus 1-2-3. Nonetheless, Borland's programs continued to be partially compatible with Lotus 1-2-3, for Borland retained what it called the "Key Reader" in its Quattro Pro programs. Once turned on, the Key Reader allowed Borland's programs to understand and perform some Lotus 1-2-3 macros.[3] With the Key Reader on, the Borland pro grams used Quattro Pro menus for display, interaction, and macro execution, except when they encountered a slash ("/") key in a macro (the starting key for any Lotus 1-2-3 macro), in which case they interpreted the macro as having been written for Lotus 1-2-3. Accordingly, people who wrote or purchased macros to shorten the time needed to perform an operation in Lotus 1-2-3, could still use those macros in Borland's programs.[4] The district court permitted Lotus to file a supplemental complaint alleging that the Key Reader infringed its copyright.

The parties agreed to try the remaining liability issues without a jury. The district court held two trials, the Phase I trial covering all remaining issues raised in the original complaint (relating to the Emulation Interface) and the Phase II trial covering all issues raised in the supplemental complaint (relating to the Key Reader). At the Phase I trial, there were no live witnesses, although considerable testimony was presented in the form of affidavits and deposition excerpts. The district court ruled upon evidentiary objections counsel interposed. At the Phase II trial, there were two live witnesses,

[3] Because Borland's programs could no longer display the Lotus menu command hierarchy to users, the Key Reader did not allow debugging or modification of macros, nor did it permit the execution of most interactive macros.

[4] *See Borland IV*, 831 F.Supp. at 226-27, for a more detailed explanation of macros and the Key Reader.

each of whom demonstrated the programs for the district court.

After the close of the Phase I trial, the district court permitted Borland to amend its answer to include the affirmative defense of "fair use." Because Borland had presented all of the evidence supporting its fair-use defense during the Phase I trial, but Lotus had not presented any evidence on fair use (as the defense had not been raised before the conclusion of the Phase I trial), the district court considered Lotus's motion for judgment on partial findings of fact. *See* Fed.R.Civ.P. 52(c). The district court held that Borland had failed to show that its use of the Lotus 1-2-3 menu command hierarchy in its Emulation Interface was a fair use. *See Borland III*, 831 F.Supp. at 208.

In its Phase I-trial decision, the district court found that "each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression." *Borland III*, 831 F.Supp. at 218. The district court also rejected Borland's affirmative defenses of laches and estoppel. *Id.* at 218–23.

In its Phase II-trial decision, the district court found that Borland's Key Reader file included "a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names." *Borland IV*, 831 F.Supp. at 228. In other words, Borland's programs no longer included the Lotus command terms, but only their first letters. The district court held that "the Lotus menu structure, organization, and first letters of the command names . . . constitute part of the protectable expression found in [Lotus 1-2-3]." *Id.* at 233. Accordingly, the district court held that with its Key Reader, Borland had infringed Lotus's copyright. *Id.* at 245. The district court also rejected Borland's affirmative defenses of waiver, laches, estoppel, and fair use. *Id.* at 235-45. The district court then entered a permanent injunction against Borland, *id.* at 245, from which Borland appeals.

This appeal concerns only Borland's copying of the Lotus menu command hierarchy into its Quattro programs and Borland's affirmative defenses to such copying. Lotus has not

cross-appealed; in other words, Lotus does not contend on appeal that the district court erred in finding that Borland had not copied other elements of Lotus 1-2-3, such as its screen displays.

II.

Discussion

On appeal, Borland does not dispute that it factually copied the words and arrangement of the Lotus menu command hierarchy. Rather, Borland argues that it "lawfully copied the unprotectable menus of Lotus 1-2-3." Borland contends that the Lotus menu command hierarchy is not copyrightable because it is a system, method of operation, process, or procedure foreclosed from protection by 17 U.S.C. § 102(b). Borland also raises a number of affirmative defenses.

A. Copyright Infringement Generally

To establish copyright infringement, a plaintiff must prove "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361, 111 S.Ct. 1282, 1296, 113 L.Ed.2d 358 (1991); *see also Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1160 n. 19 (1st Cir.1994); *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 605 (1st Cir.1988). To show ownership of a valid copyright and therefore satisfy *Feist*'s first prong, a plaintiff must prove that the work as a whole is original and that the plaintiff complied with applicable statutory formalities. *See Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1340 (5th Cir.1994). "In judicial proceedings, a certificate of copyright registration constitutes *prima facie* evidence of copyrightability and shifts the burden to the defendant to demonstrate why the copyright is not valid." *Bibbero Sys., Inc. v. Colwell Sys., Inc.*, 893 F.2d 1104, 1106 (9th Cir.1990); *see also* 17 U.S.C. § 410(c); *Folio Impressions, Inc. v. Byer California*, 937 F.2d 759, 763 (2d Cir.1991) (presumption of validity may be rebutted).

To show actionable copying and therefore satisfy *Feist's* second prong, a plaintiff must first prove that the alleged infringer copied plaintiff's copyrighted work as a factual matter; to do this, he or she may either present direct evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity). *Engineering Dynamics*, 26 F.3d at 1340; *see also Concrete Mach.*, 843 F.2d at 606. The plaintiff must then prove that the copying of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar. *See Engineering Dynamics*, 26 F.3d at 1341.

In this appeal, we are faced only with whether the Lotus menu command hierarchy is copyrightable subject matter in the first instance, for Borland concedes that Lotus has a valid copyright in Lotus 1-2-3 as a whole[5] and admits to factually copying the Lotus menu command hierarchy. As a result, this appeal is in a very different posture from most copyright-infringement cases, for copyright infringement generally turns or whether the defendant has copied protected expression as a factual matter. Because of this different posture, most copyright-infringement cases provide only limited help to us in deciding this appeal. This is true even with respect to those copyright-infringement cases that deal with computers and computer software.

---

[5] Computer programs receive copyright protection as "literary works." *See* 17 U.S.C. § 102(a)(1) (granting protection to "literary works") *and* 17 U.S.C. § 101 (defining "literary works" as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, phonorecords, film, tapes, *disks*, or cards, in which they are embodied" (emphasis added)); *see also* H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.").

B. Matter of First Impression

Whether a computer menu command hierarchy constitutes copyrightable subject matter is a matter of first impression in this court. While some other courts appear to have touched on it briefly in dicta, *see, e.g., Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 n. 23 (10th Cir.), *cert. denied*, ___ U.S. ___, 114 S.Ct. 307, 126 L.Ed.2d 254 (1993), we know of no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue). Thus we are navigating in uncharted waters.

Borland vigorously argues, however, that the Supreme Court charted our course more than 100 years ago when it decided *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879). In *Baker v. Selden*, the Court held that Selden's copyright over the textbook in which he explained his new way to do accounting did not grant him a monopoly on the use of his accounting system.[6] Borland argues:

> The facts of *Baker v. Selden*, and even the arguments advanced by the parties in that case, are identical to those in this case. The only difference is that the "user interface" of Selden's system was implemented by pen and paper rather than by computer.

To demonstrate that *Baker v. Selden* and this appeal both involve accounting systems, Borland even supplied this court with a video that, with special effects, shows Selden's paper forms "melting" into a computer screen and transforming into Lotus 1-2-3.

We do not think that *Baker v. Selden is* nearly as analogous to this appeal as Borland claims. Of course, Lotus 1-2-3 is a computer spreadsheet, and as such its grid of horizontal rows and vertical columns certainly resembles an accounting ledger or any other paper spreadsheet. Those grids, however, are not at issue in this appeal for, unlike Selden, Lotus does not claim to have a monopoly over its accounting system. Rather, this

[6] Selden's system of double-entry bookkeeping is the now almost-universal T-accounts system.

appeal involves Lotus's monopoly over the commands it uses to operate the computer. Accordingly, this appeal is not, as Borland contends, "identical" to *Baker v. Selden*.

C. Altai

Before we analyze whether the Lotus menu command hierarchy is a system, method of operation, process, or procedure, we first consider the applicability of the test the Second Circuit set forth in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir.1992).[7] The Second Circuit designed its *Altai* test to deal with the fact that computer programs, copyrighted as "literary works," can be infringed by what is known as "nonliteral" copying, which is copying that is paraphrased or loosely paraphrased rather than word for word. *See id.* at 701 (citing nonliteral-copying cases); *see also* 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A][1] (1993). When faced with nonliteral-copying cases, courts must determine whether similarities are due merely to the fact that the two works share the same underlying idea or whether they instead indicate that the second author copied the first author's expression. The Second Circuit designed its *Altai* test to deal with this situation in the computer context, specifically with whether one computer program copied nonliteral expression from another program's code.

The *Altai* test involves three steps: abstraction, filtration, and comparison. The abstraction step requires courts to "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it." *Altai* 982 F.2d at 707. This step enables courts to identify the appropriate framework within which to separate protectable expression from unprotected ideas. Second, courts apply a "filtration"

[7] We consider the *Altai* test because both parties and many of the *amici* focus on it so heavily. Borland, in particular, is highly critical of the district court for not employing the *Altai* test. Borland does not, however, indicate how using that test would have been dispositive in Borland's favor. Interestingly, Borland appears to contradict its own reasoning at times by criticizing the applicability of the *Altai* test.

step in which they examine "the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain." *Id.* Finally, courts compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement. *Id.* at 710.

In the instant appeal, we are not confronted with alleged nonliteral copying of computer code. Rather, we are faced with Borland's deliberate, literal copying of the Lotus menu command hierarchy. Thus, we must determine not whether nonliteral copying occurred in some amorphous sense, but rather whether the literal copying of the Lotus menu command hierarchy constitutes copyright infringement.

While the *Altai* test may provide a useful framework for assessing the alleged nonliteral copying of computer code, we find it to be of little help in assessing whether the literal copying of a menu command hierarchy constitutes copyright infringement. In fact, we think that the *Altai* test in this context may actually be misleading because, in instructing courts to abstract the various levels, it seems to encourage them to find a base level that includes copyrightable subject matter that, if literally copied, would make the copier liable for copyright infringement.[8] While that base (or literal) level would not be at issue in a nonliteral-copying case like *Altai*, it is precisely what is at issue in this appeal. We think that abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from

[8] We recognize that *Altai* never states that every work contains a copyrightable "nugget" of protectable expression. Nonetheless, the implication is that for literal copying, "it is not necessary to determine the level of abstraction at which similarity ceases to consist of an 'expression of ideas,' because literal similarity by definition is always a similarity as to the expression old ideas." 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A](2) (1993).

expression at that stage, as both the *Altai* and the district court tests require, obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all. The initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but rather whether the menu command hierarchy as a whole can be copyrighted. *But see Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823 (10th Cir.1993) (endorsing *Altai*'s abstraction-filtration-comparison test as a way of determining whether "menus and sorting criteria" are copyrightable).

D. The Lotus Menu Command Hierarchy: A "Method of Operation"

Borland argues that the Lotus menu command hierarchy is uncopyrightable because it is a system, method of operation, process, or procedure foreclosed from copyright protection by 17 U.S.C. § 102(b). Section 102(b) states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." Because we conclude that the Lotus menu command hierarchy is a method of operation, we do not consider whether it could also be a system, process, or procedure.

We think that "method of operation," as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer. Thus a text describing how to operate something would not extend copyright protection to the method of operation itself; other people would be free to employ that method and to describe it in their own words. Similarly, if a new method of operation is used rather than described, other people would still be free to employ or describe that method.

We hold that the Lotus menu command hierarchy is an uncopyrightable "method of operation." The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use

the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities.

The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled. The Lotus menu command hierarchy is different from the Lotus long prompts, for the long prompts are not necessary to the operation of the program; users could operate Lotus 1-2-3 even if there were no long prompts.[9] The Lotus menu command hierarchy is also different from the Lotus screen displays, for users need not "use" any expressive aspects of the screen displays in order to operate Lotus 1-2-3; because the way the screens look has little bearing on how users control the program, the screen displays are not part of Lotus 1-2-3's "method of operation."[10] The Lotus menu command hierarchy is also different from the underlying computer code, because while code is necessary for the program to work, its precise formulation is not. In other words, to offer the same capabilities as Lotus 1-2-3, Borland did not have to copy Lotus's underlying code (and indeed it did not); to allow users to operate its programs in substantially the same way, however, Borland had to copy the Lotus menu command hierarchy. Thus the Lotus 1-2-3 code is not a uncopyrightable "method of operation."[11]

---

[9] As the Lotus long prompts are not before us appeal, we take no position on their copyrightability, although we do note that a strong argument could be made that the brief explanations they provide "merge" with the underlying idea of explaining such functions. *See Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir.1967) (when the possible ways to express an idea are limited, the expression "merges" with the idea and is therefore uncopyrightable; when merger occurs, identical copying is permitted).

[10] As they are not before us on appeal, we take no position on whether the Lotus 1-2-3 screen displays constitute original expression capable of being copyrighted.

[11] Because the Lotus 1-2-3 code is not before us on appeal, we take no position on whether it is copyrightable. We note, however, that original computer codes generally are protected by copyright. *See, e.g., Altai*, 982 F.2d at 702 ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.") (citing cases).

The district court held that the Lotus menu command hierarchy, with its specific choice and arrangement of command terms, constituted an "expression" of the "idea" of operating a computer program with commands arranged hierarchically into menus and submenus. *Borland II*, 799 F.Supp. at 216. Under the district court's reasoning, Lotus's decision to employ hierarchically arranged command terms to operate its program could not foreclose its competitors from also employing hierarchically arranged command terms to operate their programs, but it did foreclose them from employing the specific command terms and arrangement that Lotus had used. In effect, the district court limited Lotus 1-2-3's "method of operation" to an abstraction.

Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that that expression is not copyrightable because it is part of Lotus 1-2-3's "method of operation." We do not think that "methods of operation" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable. This is so whether they must be highlighted, typed in, or even spoken, as computer programs no doubt will soon be controlled by spoken words.

The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." In other words, our initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression.[12] Rather, our initial inquiry is whether the Lotus menu command hierarchy is a "method of operation." Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hier-

---

[12] We think that the *Altai* test would contemplate this being the initial inquiry.

archy, and that the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.

Our holding that "methods of operation" are not limited to mere abstractions is bolstered by *Baker v. Selden*. In *Baker*, the Supreme Court explained that

> the teachings of science and the rules and methods of useful art have their final end in application and use; and this application and use are what the public derive from the publication of a book which teaches them. . . . The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.

Baker v. Selden, 101 U.S. at 104-05. Lotus wrote its menu command hierarchy so that people could learn it and use it. Accordingly, it falls squarely within the prohibition on copyright protection established in *Baker v. Selden* and codified by Congress in § 102(b).

In many ways, the Lotus menu command hierarchy is like the buttons used to control, say, a video cassette recorder ("VCR"). A VCR is a machine that enables one to watch and record video tapes. Users operate VCRs by pressing a series of buttons that are typically labelled "Record, Play, Reverse, Fast Forward, Pause, Stop/Eject." That the buttons are arranged and labeled does not make them a "literary work," nor does it make them an "expression" of the abstract "method of operating" a VCR via a set of labeled buttons. Instead, the buttons are themselves the "method of operating" the VCR.

When a Lotus 1-2-3 user chooses a command, either by highlighting it on the screen or by typing its first letter, he or

she effectively pushes a button. Highlighting the "Print" command on the screen, or typing the letter "P," is analogous to pressing a VCR button labeled "Play."

Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. Unlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are essential to operating Lotus 1-2-3. Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons. While Lotus could probably have designed a user interface for which the command terms were mere labels, it did not do so here. Lotus 1-2-3 depends for its operation on use of the precise command terms that make up the Lotus menu command hierarchy.

One might argue that the buttons for operating a VCR are not analogous to the command for operating a computer program because VCRs are not copyrightable, where as computer programs are. VCRs may not be copyrighted because they do not fit within any of the § 102(a) categories of copyrightable works; the closest they come is "sculptural work." Sculptural works, however, are subject to a "useful-article" exception whereby "the design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101. A "useful article" is "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." *Id.* Whatever expression there may be in the arrangement of the parts of a VCR is not capable of existing separately from the VCR itself, so an ordinary VCR would not be copyrightable.

Computer programs, unlike VCRs, are copyrightable as "literary works." 17 U.S.C. § 102(a). Accordingly, one might

argue, the "buttons" used to operate a computer program are not like the buttons used to operate a VCR, for they are not subject to a useful-article exception. The response, of course, is that the arrangement of buttons on a VCR would not be copyrightable even without a useful-article exception, because the buttons are an uncopyrightable "method of operation." Similarly, the "buttons" of a computer program are also an uncopyrightable "method of operation."

That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. For example, if the user wanted the computer to print material, then the user would have to learn not just one method of operating the computer such that it prints, but many different methods. We find this absurd. The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable.

Consider also that users employ the Lotus menu command hierarchy in writing macros. Under the district court's holding, if the user wrote a macro to shorten the time needed to perform a certain operation in Lotus 1-2-3, the user would be unable to use that macro to shorten the time needed to perform that same operation in another program. Rather, the user would have to rewrite his or her macro using that other program's menu command hierarchy. This is despite the fact that the macro is clearly the user's own work product. We think that forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that "methods of operation" are not copyrightable. That programs can offer users the ability to write macros in many different ways does not change the fact that, once written, the macro allows the user to perform an operation automatically. As the Lotus menu command hierarchy

serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a "method of operation."

In holding that expression that is part of a "method of operation" cannot be copyrighted, we do not understand ourselves to go against the Supreme Court's holding in *Feist*. In *Feist*, the Court explained:

> The primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work.

Feist, 499 U.S. at 349-50, 111 S.Ct. at 1290 (quotations and citations omitted). We do not think that the Court's statement that "copyright assures authors the right to their original expression" indicates that all expression is necessarily copyrightable; while original expression is necessary for copyright protection, we do not think that it is alone sufficient. Courts must still inquire whether original expression falls within one of the categories foreclosed from copyright protection by § 102(b), such as being a "method of operation."

We also note that in most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression.[13] In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Our holding that methods of operation are not limited to abstractions goes against *Autoskill*, 994 F.2d at 1495 n. 23, in which the Tenth Circuit rejected the defendant's argument that

---

[13] When there are a limited number of ways to express an idea, however, the expression "merges" with the idea and becomes uncopyrightable. *Morissey*, 379 F.2d at 678-79.

the keying procedure used in a computer program was an uncopyrightable "procedure" or "method of operation" under § 102(b). The program at issue, which was designed to test and train students with reading deficiencies, *id.* at 1481, required students to select responses to the program's queries "by pressing the 1, 2, or 3 keys." *Id.* at 1495 n. 23. The Tenth Circuit held that, "for purposes of the preliminary injunction, . . . the record showed that [this] keying procedure reflected at least a minimal degree of creativity," as required by *Feist* for copyright protection. *Id.* As an initial matter, we question whether a programmer's decision to have users select a response by pressing the 1, 2, or 3 keys is original. More importantly, however, we fail to see how "a student select[ing] a response by pressing the 1, 2, or 3 keys," *id.*, can be anything but an unprotectable method of operation.[14]

III.

Conclusion

Because we hold that the Lotus menu command hierarchy is uncopyrightable subject matter, we further hold that Borland did not infringe Lotus's copyright by copying it. Accordingly, we need not consider any of Borland's affirmative defenses. The judgment of the district court is

Reversed.

Concurrence follows.

---

[14] The Ninth Circuit has also indicated in dicta that "menus, and keystrokes" may be copyrightable. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1477 (9th Cir.), *cert. denied, BB Asset Management, Inc. v. Symantec Corp.*, ___ U.S. ___, 113 S.Ct. 198, 121 L.Ed.2d 141 (1992). In that case, however, the plaintiff did not show that the defendant had copied the plaintiff's menus or keystrokes, so the court was not directly faced with whether the menus or keystrokes constituted an unprotectable method of operation. *Id.*

BOUDIN, *Circuit Judge*, concurring.

The importance of this case, and a slightly different emphasis in my view of the underlying problem, prompt me to add a few words to the majority's tightly focused discussion.

I.

Most of the law of copyright and the "tools" of analysis have developed in the context of literary works such as novels, plays, and films. In this milieu, the principal problem—simply stated, if difficult to resolve—is to stimulate creative expression without unduly limiting access by others to the broader themes and concepts deployed by the author. The middle of the spectrum presents close cases; but a "mistake" in providing too much protection involves a small cost: subsequent authors treating the same themes must take a few more steps away from the original expression.

The problem presented by computer programs is fundamentally different in one respect. The computer program is a *means* for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work. Granting protection, in other words, can have some of the consequences of *patent* protection in limiting other people's ability to perform a task in the most efficient manner. Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.

Of course, the argument *for* protection is undiminished, perhaps even enhanced, by utility: if we want more of an intellectual product, a temporary monopoly for the creator provides incentives for others to create other, different items in this class. But the "cost" side of the equation may be different where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task. Thus, the argument for extending protection may be the same; but the stakes on the other side are much higher.

It is no accident that patent protection has preconditions that copyright protection does not—notably, the requirements

of novelty and non-obviousness—and that patents are granted for a shorter period than copyrights. This problem of utility has sometimes manifested itself in copyright cases, such as *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), and been dealt with through various formulations that limit copyright or create limited rights to copy. But the case law and doctrine addressed to utility in copyright have been brief detours in the general march of copyright law.

Requests for the protection of computer menus present the concern with fencing off access to the commons in an acute form. A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by *users* in learning the menu and in building their own mini-programs—macros—in reliance upon the menu. Better typewriter keyboard layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has learned to use. *See* P. David, *CLIO and the Economics of QWERTY*, 75 *Am.Econ.Rev.* 332 (1985). The QWERTY keyboard is nothing other than a menu of letters.

Thus, to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. The "form"—the written source code or the menu structure depicted on the screen—look hauntingly like the familiar stuff of copyright; but the "substance" probably has more to do with problems presented in patent law or, as already noted, in those rare cases where copyright law has confronted industrially useful expressions. Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.

All of this would make no difference if Congress had squarely confronted the issue, and given explicit directions as to what should be done. The Copyright Act of 1976 took a different course. While Congress said that computer programs might be subject to copyright protection, it said this in very general terms; and, especially in § 102(b), Congress adopted a string of exclusions that if taken literally might easily seem to exclude most computer programs from protection. The only

detailed prescriptions for computers involve narrow issues (like back-up copies) of no relevance here.

Of course, one could still read the statute as a congressional command that the familiar doctrines of copyright law be taken and applied to computer programs, in cookie cutter fashion, as if the programs were novels or play scripts. Some of the cases involving computer programs embody this approach. It seems to be mistaken on two different grounds: the tradition of copyright law, and the likely intent of Congress.

The broad-brush conception of copyright protection, the time limits, and the formalities have long been prescribed by statute. But the heart of copyright doctrine—what may be protected and with what limitations and exceptions—has been developed by the courts through experience with individual cases. B. Kaplan, *An Unhurried View of Copyright* 40 (1967). Occasionally Congress addresses a problem in detail. For the most part the interstitial development of copyright through the courts is our tradition.

Nothing in the language or legislative history of the 1976 Act, or at least nothing brought to our attention, suggests that Congress meant the courts to abandon this case-by-case approach. Indeed, by setting up § 102(b) as a counterpoint theme, Congress has arguably recognized the tension and left it for the courts to resolve through the development of case law. And case law development is *adaptive:* it allows new problems to be solved with help of earlier doctrine, but it does not preclude new doctrines to meet new situations.

II.

In this case, the raw facts are mostly, if not entirely, undisputed. Although the inferences to be drawn may be more debatable, it is very hard to see that Borland has shown any interest in the Lotus menu except as a fall-back option for those users already committed to it by prior experience or in order to run their own macros using 1-2-3 commands. At least for the amateur, accessing the Lotus menu in the Borland Quattro or Quattro Pro program takes some effort.

Put differently, it is unlikely that users who value the Lotus menu for its own sake—independent of any investment they have made themselves in learning Lotus' commands or creating macros dependent upon them—would choose the Borland program in order to secure access to the Lotus menu. Borland's success is due primarily to other features. Its rationale for deploying the Lotus menu bears the ring of truth.

Now, any use of the Lotus menu by Borland is a commercial use and deprives Lotus of a portion of its "reward," in the sense that an infringement claim if allowed would increase Lotus' profits. But this is circular reasoning: broadly speaking, every limitation on copyright or privileged use diminishes the reward of the original creator. Yet not every writing is copyrightable or every use an infringement. The provision of reward is one concern of copyright law, but it is not the only one. If it were, copyrights would be perpetual and there would be no exceptions.

The present case is an unattractive one for copyright protection of the menu. The menu commands (*e.g.*, "print," "quit") are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize. What is left is the particular combination and sub-grouping of commands in a pattern devised by Lotus. This arrangement may have a more appealing logic and ease of use than some other configurations; but there is a certain arbitrariness to many of the choices.

If Lotus is granted a monopoly on this pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus, just as a typist who has learned the QWERTY keyboard would be the captive of anyone who had a monopoly on the production of such a keyboard. Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the *de facto* standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet—either in quality or in price—there may be nothing wrong with this advantage.

But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an

investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance, and to reward Borland in turn for making a better product. If Borland has not made a better product, then customers will remain with Lotus anyway.

Thus, for me the question is not whether Borland should prevail but on what basis. Various avenues might be traveled, but the main choices are between holding that the menu is not protectable by copyright and devising a new doctrine that Borland's use is privileged. No solution is perfect and no intermediate appellate court can make the final choice.

To call the menu a "method of operation" is, in the common use of those words, a defensible position. After all, the purpose of the menu is not to be admired as a work of literary or pictorial art. It is to transmit directions from the user to the computer, *i.e., to operate* the computer. The menu is also a "method" in the dictionary sense because it is a "planned way of doing something," an "order or system," and (aptly here) an "orderly or systematic arrangement, sequence or the like." *Random House Webster's College Dictionary* 853 (1991).

A different approach would be to say that Borland's use is privileged because, in the context already described, it is not seeking to appropriate the advances made by Lotus' menu; rather, having provided an arguably more attractive menu of its own, Borland is merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros. The difference is that such a privileged use approach would not automatically protect Borland if it had simply copied the Lotus menu (using different codes), contributed nothing of its own, and resold Lotus under the Borland label.

The closest analogue in conventional copyright is the fair use doctrine. *E.g., Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 105 S.Ct. 2218, 85 L.Ed.2d 588 (1985). Although invoked by Borland, it has largely been brushed aside in this case because the Supreme Court has said

that it is "presumptively" unavailable where the use is a "commercial" one. *See id.* at 562, 105 S.Ct. at 2231-32. *But see Campbell v. Acuff-Rose Music, Inc.*, ___ U.S. ___, ___, 114 S.Ct. 1164, 1174, 127 L.Ed.2d 500 (1994). In my view, this is something less than a definitive answer; "presumptively" does not mean "always" and, in any event, the doctrine of fair use was created by the courts and can be adapted to new purposes.

But a privileged use doctrine would certainly involve problems of its own. It might more closely tailor the limits on copyright protection to the reasons for limiting that protection; but it would entail a host of administrative problems that would cause cost and delay, and would also reduce the ability of the industry to predict outcomes. Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.

In sum, the majority's result persuades me and its formulation is as good, if not better, than any other that occurs to me now as within the reach of courts. Some solutions (*e.g.*, a very short copyright period for menus) are not options at all for courts but might be for Congress. In all events, the choices are important ones of policy, not linguistics, and they should be made with the underlying considerations in view.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K
Aug. 12, 1993.
As Amended Aug. 19, 1993.
Permanent Injunction Aug. 19, 1993.

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

KEETON, District Judge.

A nonjury trial was held in this action in two phases. Phase I was tried on February 1–3, 1993; Phase II, on March 31–April 2, 1993. An Opinion of June 30, 1993 (the "Phase I Opinion") stated the court's findings and conclusions on issues raised in Phase I of the trial. The present Opinion states my findings and conclusions for the Phase II trial.

I. Introduction.

This Opinion assumes the reader's familiarity with, and follows the terminology set forth in, the Phase I Opinion. Background information appears also in *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) (*"Borland II"*), *Lotus Dev. Corp. v. Borland Int'l Inc.*, 788 F.Supp. 78 (D.Mass.1992) (*"Borland I"*), and *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990) (*"Paperback"*).

As explained in the Phase I Opinion, the Phase I trial concerned issues of infringement raised in Lotus's original complaint, *ie.*, infringement by Borland's copying of Lotus's menu tree into the 1-2-3 emulation interface of Borland's Quattro and certain Quattro Pro spreadsheet programs.

Phase II of the trial concerns the full range of liability issues, including affirmative defenses, with respect to Lotus's supplemental complaint. The supplemental complaint alleges that Borland's "Key Reader" infringes Lotus's copyrights in its 1-2-3 programs.

II. Does Key Reader Infringe?

A. What is Key Reader?

The Phase I Opinion and the court's earlier Memorandum and Orders have focused on the 1-2-3 emulation interfaces in the Quattro and Quattro Pro programs. After executing the command that invokes the emulation interface in one of Borland's products, the program uses copies of the Lotus menu

tree for display to the user, interaction with the user (*i.e.*, interpreting the keystrokes typed by the user), and running macros.

The Key Reader is not a part of the emulation interface. The Key Reader feature may be turned on while the user continues to use another (*e.g.*, the Quattro) menu tree. When Key Reader has been turned on, the program continues to behave as it had before, with one exception. With Key Reader on, when the program encounters a slash key ("/") in a macro, the program interprets everything that follows the slash key as though it were part of a macro written for use with Lotus 1-2-3. Thus with Key Reader on, the program uses Quattro Pro menus for display, interaction and macro execution, except when a "/" key is encountered in a macro. Then, the program runs macros as though they were written using the 1-2-3 menu tree.

A more detailed explanation of 1-2-3 "macros" is essential to explaining Key Reader.

Generally, a 1-2-3 macro consists of some text contained in a spreadsheet cell. When the macro is invoked, the program begins at the left end of the text and treats each character in the text as though the user had typed that character into the 1-2-3 program. For example, consider a macro that consists of the text "/wp". When the macro is invoked, the program behaves precisely as though the user had typed into the regular interface a "/" (calling up the first menu) followed by "w" (selecting the "Worksheet" branch of the menu tree) followed by "p" (selecting the "Page" leaf; this executable operation inserts a page break into the spreadsheet).

In writing a macro, the user may use special commands not found in the 1-2-3 menu tree. For example, the character "~" in a macro is interpreted as though the user struck the Enter key on the keyboard. Consider a macro consisting of the text "/rfc~~". When invoked, this macro has the same effect as the user's typing into the 1-2-3 interface "/", "r", "f", "c", followed by striking the Enter key twice. That is, the program follows the "Range" branch, then the "Format" branch, then selects the "Currency" leaf. When the "Currency" leaf has

been selected, the program asks the user to specify the number of decimal places and the block of cells whose appearance is to be altered to that of monetary units. The first "~" is equivalent to striking the Enter key; the program accepts the default number of decimal places (two). The program then asks for the range of cells. The second "~" is equivalent to striking the Enter key a second time; the program accepts the default range (one cell).

Other sequences of symbols in a Lotus macro have a special meaning. Of particular interest is "{?}". When the program encounters this sequence of characters in a macro, the program pauses and waits for the user to strike keys appropriate for the menu tree. Thus, consider a Lotus macro consisting of the text "/rf{?}~~". When invoked, this macro would follow the "r" command ("Range" from the first level of the Lotus menu tree), then the "f" command ("Format" in the Lotus menu tree). The program then encounters the "{?}" characters. Here, the program pauses and allows the user to input a menu command. The program will interpret what the user inputs as a choice from the Lotus 1-2-3 submenu corresponding to "Range" "Format". Thus, the user might strike the "c" key (followed by Enter) to format a cell as currency. If the user strikes the Escape key instead of striking "c", the program backs up the menu tree from the "Format" submenu to the "Range" submenu. The user may continue maneuvering up and down the Lotus 1-2-3 menu tree until striking the Enter key. When the user strikes the Enter key, the program returns to reading the text of the macro as though the user were continuing by striking the keys found in the remaining text of the macro. A macro consisting only of "/{?}" permits a user to select any executable operation by typing keys precisely as though the user were using one of Borland's emulation interfaces, but without display of the menus.

Lotus 1-2-3 macros may also contain more advanced commands. Thus, a macro may contain text such as "{if . . .}" or "{let . . .}". In these strings of characters, the ellipses refer to additional text or "arguments" the user would provide. When the program encounters an advanced command such as

this, the program follows whatever procedures that command calls for. For example, the text "{blank A1..G45}" would cause the program to erase each of the cells from "A1" (first column and first row) through "G45" (seventh column and forty-fifth row). "if", "let", and "blank" are all labels in the Lotus 1-2-3 macro language that do not appear in the 1-2-3 menu tree.

In sum, when Key Reader is on and a slash key is encountered in a macro, the program follows the text of the macro as though the characters were being typed during the program's use of a copy of the 1-2-3 menu tree (and with some other means for interpreting the various special Lotus 1-2-3 macro language commands). When Key Reader is off, or no "/" key is encountered in a macro, the program interprets macros by reference to a different (*e.g.*, the Quattro) menu tree. With Key Reader off, Borland's programs cannot correctly interpret 1-2-3 macros.

Quattro and Quattro Pro version 1.0 did not contain the Key Reader feature; they contained only the emulation interface. Thus in these programs, the user could execute Lotus macros only when the user was employing the emulation interface; *ie.*, using Lotus 1-2-3 menus for display, interpreting user commands, and executing macros.

The Key Reader was first introduced in Quattro Pro version 2.0. Quattro Pro version 2.0, 3.0, and 4.0 contain *both* the emulation interface and the Key Reader feature (which is used from the native menus).

Borland removed the emulation interface from Quattro Pro version 4.01 (and subsequent releases of Quattro Pro) after the *Borland II* decision in this case allowed partial summary judgment for Lotus. Thus, Quattro Pro versions 4.01, SE, and Quattro Pro for Windows contain the Key Reader feature but have no 1-2-3 emulation interface.

B. Copying.

In developing Key Reader, Borland modified portions of the earlier Quattro Pro programs that contained the full 1-2-3

emulation interface. Accordingly, I begin with a review of how the Lotus menu tree was copied into the Quattro Pro emulation interface before analyzing how that was, in turn, incorporated into Key Reader.

1. The 123.MU file.

The actual menu tree for the Quattro Pro emulation interfaces (versions 1.0, 2.0, 3.0, and 4.0) was in a file labeled "123.MU". Thus, when the program was running and the user had selected the 1-2-3 emulation interface, the program would refer to the 123.MU file to determine the form of menu commands and menu structure used to present the set of executable operations to the user, interpret user commands, and interpret macros. If the user had selected a different interface, *e.g.*, one using the Quattro native menus, the program would refer to a different file for the menu commands and menu structure for display, interpretation of commands, and macro execution.

A text print-out of portions of the 123.MU file is in evidence under seal as Exhibit 13. The parties have included, however, the first page of this print-out in the public record. *See* Docket 311, Exh. A. An examination of the print-out demonstrates that the entire Lotus menu tree is copied into the file, with differences in indentation detailing the menu structure. That is, the structure of the menus and submenus is recorded in the file by changes in indentation. As explained in the Opinion for Phase I, when the program is using the 123.MU file as the source for its menus, it presents to the user a virtually identical copy of the Lotus menu tree (with Borland's additional menu commands inserted into the tree).

2. Key Reader files.

As the above explanation of macros suggests, interpretation of Lotus macros requires that the program treat the characters in the macro text as though they were keystrokes into a spreadsheet program using the Lotus menu tree (except for the characters in the macro text that are special commands in the Lotus macro language).

I find that, to implement the Key Reader (or at least that portion of the Key Reader that interprets characters from the Lotus menu tree), Borland began with the 123.MU file from the emulation interface. Borland then prepared a new file by reproducing the old 123.MU file but with only the first letter of each menu command name where the entire Lotus menu command name appears in the old 123.MU file. *See* Warfield Dep'n. IX:43. Put another way, the point is that to implement Key Reader Borland used a program file containing the same copy of the 1-2-3 menu tree structure and commands that Borland had used in its emulation interface, but with each menu command name stripped of everything after the first letter. Borland then appended this copy of the "stripped menu tree" to its quattro.mu file.

At various stages, Borland introduced other changes in how the menu structure is recorded in the stripped file, altering parentheses or indentation, etc. For example, I find that Borland may have altered the symbols used to record Lotus's menu structure, inserted the word "PICKLETTER" before the first letter of the menu command, or made other changes. *Compare* Exhibit 13 *with* Exhibit 517; *see* Warfield Dep'n, IX:43 at 148-49. Some of these things may actually appear in the old 123.MU file—Exhibit 13 contains only portions of the old 123.MU file. *See* Warfield Dep'n, IX:43 at 107-109, 148-49. In any event, I have considered the file and possible changes. I find that none of these changes is material to the scope or nature of copying from the Lotus 1-2-3 program.

In short, I find that the Key Reader file contains a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names.

Borland contends that the command letters copied from Lotus 1-2-3 that are in Borland's Key Reader files are not in the same order as displayed on the screen in Lotus 1-2-3. That is, if one reads sequentially down the Key Reader file, one encounters the Lotus commands in a different order than when reading the menus from the display screen while running Lotus 1-2-3. This is true, however, only in the sense that

the menu structure of Lotus 1-2-3 is represented in a different way in the Key Reader *file* than on the Lotus 1-2-3 *display screens;* in the file, the structure is detailed by differences in indentation (or other means) rather than through display on the screen. I find that the file (in evidence under seal, Exhibit 517) fully delineates a virtually identical copy of the menu structure of Lotus 1-2-3 including the first letter of each menu command in the corresponding location in the copy of the menu structure.

Because these menus are used in interpreting macros but are never fully displayed to the user, the parties have sometimes referred to these as "phantom menus."

In sum, to interpret macros, Borland's programs use a file with phantom menus consisting of a virtually identical copy of the Lotus menu tree that Borland used for its emulation interface, but with only the first letter of each menu command name where the complete menu command name previously appeared.

C. Copyrightabilitiy Issues.

1. Renewed and new arguments concerning definition of the idea.

Borland contends that copyright protection covering the structure of the 1-2-3 menu tree and the first letters of the commands in the 1-2-3 menu tree *(ie.,* Borland's phantom menus) would be equivalent to copyright protection for a "system" or "method" of communication between the user and the program. Thus, Borland argues that copying of the 1-2-3 menu tree structure and first letters of command names is a necessary part of any *system* for interpreting Lotus 1-2-3 macros. Citing the proposition that copyright law does not protect a "system", *see* 17 U.S.C. § 102(b), Borland argues that the Lotus copyright cannot extend to Borland's phantom menus.

Lotus responds that Borland has not proved that copying of any part of the menu tree is necessary for running or translating macros.

As a preliminary matter, I observe that the parties, witnesses (and the court, in some instances) have not used the terms "macro translation," "macro conversion," "macro execution," and "macro compatibility" with precision during the Phase II trial and earlier proceedings. In order to avoid possible future misunderstanding, I will explain the ambiguity in use of these terms before proceeding farther.

Macro "conversion" refers to translating a macro that was written using Lotus 1-2-3 into a macro written in a form for use in another program. For example, the macro "/rfc~~" in Lotus may be translated, using the macro translation assistant in Excel 2.1, into an Excel 2.1 macro having the text:

```
= SET.NAME("Selection—Save", SELECTION( ))
= SET.NAME("Range3", SELECTION( ))
= DISPLAY(FALSE)
= SELECT(Range3)
= FORMAT.NUMBER
("$#,##0.00;($#,##0.00)"))
= RETURN( )
```

This macro text is written in the Excel macro language and is executed by reference to the Excel menus. Thus, formatting a single cell to be displayed in monetary units with two decimal places (the result of the "/rfc~~" macro in Lotus) may be achieved in Excel 2.1 by selecting a single cell in an Excel spreadsheet followed by selection of the "Format" menu command, the "Number" command in the submenu of Format, then the "$#,##0.00;($#,##0.00)" (currency) leaf (compare with the second to last line of the Excel macro).

This facility for conversion of macros may also be referred to as a "one-time macro translator." The translator takes a Lotus macro and converts it into a macro written in a different macro language, for use with a different menu tree. Once the translation has been made, a user may run the translated macro as frequently as desired in the other program. Because the macro is written in a different macro language, the program need not refer to a copy of Lotus's menu tree to run the (translated) macro. Thus in *Borland II*, I observed that:

> I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3. Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland's command hierarchy, that would have presented a different case from the one now before me.

799 F.Supp. at 214. In other words, I did not decide issues concerning one-time translation of macros into a different macro language, such that the macro could be executed, modified, and debugged without reference to copies of the Lotus menu tree.

Alternatively, consider a continuous macro "interpreter." This facility executes Lotus macros by referring to copies of the Lotus menu tree contained within the program. Macro interpretation may thus be viewed as continuous "translation" of macros as each macro is executed, or "on-the-fly" interpretation. Each time a macro is executed, modified, or debugged, the macro remains written in the Lotus macro language. Each time a macro is executed, modified or debugged, the program refers to the Lotus menu tree or structure.

Both one-time conversion and on-the-fly interpretation may be referred to as "macro translation," as providing an ability to "run" Lotus macros, and as providing "macro compatibility" with Lotus. For both, the user may begin with a Lotus macro and run the macro either by translating it first (one-time translation), or by parsing it with reference to the Lotus menu tree structure each time the user runs the macro (on-the-fly).

An example of the confusion in terms is demonstrated by testimony from Borland's Chief Executive Officer, Phillipe Kahn. Kahn testified that "macro translation was not a viable way to do things." I:13 at 100. From the context of this testimony, I understand the testimony to refer to one-time "macro translation" rather than what Borland now refers to as (on-the-fly) "macro translation" performed by Key Reader.

Borland's Key Reader does not perform one-time translation such as Excel 2.1 does. Rather, Key Reader interprets macros on-the-fly, by reference to "phantom" menus that contain a copy of the Lotus menu tree.

In clarifying this terminology, I do not decide whether copying of the Lotus menu structure for the purpose of one-time translation rather than on-the-fly interpretation should be accorded different treatment under copyright law. Also, I do not decide any issue concerning who (if anyone) owns the copyright in a macro written by a user in reliance on expression found in the Lotus 1-2-3 menu tree, or who (if anyone) owns the copyright in a translation into another language of a macro written in reliance on Lotus 1-2 3. These issues are not before the court. Rather, I make clear that I decide *only* those issues raised by Borland's Key Reader; *ie.*, only issues involved in on-the-fly interpretation using Borland's "phantom" menus.

Having clarified the terminology, I now reject Lotus's argument that on-the-fly macro interpretation does not require copying from the Lotus menu structure and first letters of the command names. To interpret a macro, the program must use the Lotus 1-2-3 menu structure. If a program did not have a representation of the 1-2-3 menu hierarchy somewhere within the program code (or in a file that is used by the code), then there is no way that the program could understand that "rfc" refers to a path through a menu tree to the specific executable operation that changes a cell or cells appearance to monetary units *(i.e.*, a path through the range and format menus to the currency leaf). Whether the menu tree is copied into a file or directly into the code of a program does not make a difference. The scope of copying of the menu tree is the same whether done in one computer language, a different language, or in a file. Accordingly, I reject Lotus's contention that no part of the 1-2-3 menu tree need be copied to interpret 1-2-3 macros.

Nevertheless, I also reject Borland's contention that the menu tree structure and first letters of the menu commands constitute "system" or "method," as those terms are used in copyright law.

Of course, it is possible to think of the Lotus 1-2-3 menu tree as a "system" for translating user keystrokes into executable operations (whether the keystrokes are typed into the keyboard or are taken from the text of a macro command). In the same way, a book may be thought of as a system for communicating various ideas and images to readers. Indeed, the only way of conveying each of the exact images of "Gone With the Wind" in the same order as in "Gone With the Wind" is to copy at least a great deal of the nonliteral expression in the book, if not the precise words of the book as well. Nevertheless, copyright protection extends both to the specific words of "Gone With the Wind" and to nonliteral elements of the novel. Accordingly, the ability to describe a work as a "system" is not decisive of whether the work is a "system," or instead is protected expression, under copyright law.

In calling the copying that is contained in the phantom menu tree a "system," one may attempt to distinguish the copying found in the phantom menus from copying of a book on the following grounds. A book communicates to the reader. The phantom menus, however, permit the user to communicate commands to the program. Thus, one may contend that the phantom menu trees constitute a method or system for the user to communicate executable commands rather than a system (such as a book) for communicating thoughts to the reader.

For the following reasons, I reject this basis for concluding that, in creating the phantom menus, Borland copied a system and not its expression. First, like Lotus's menu tree, the protected expression of a compilation may be viewed as a system for accessing information. The purpose of a compilation is to communicate facts. The specific facts communicated, however, are not copyrightable. Copyright protects only the selection, arrangement and manner of presentation of the facts (to the extent that those elements meet the other requirements for copyrightability). *See Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The selection, arrangement, and manner of presentation in a compilation may provide the user with a *method* or

systematic manner of accessing the (uncopyrightable) facts. Thus, copyright law protects only that part of a compilation that the reader actually *uses* for selection of facts that the reader wants to know. Nevertheless, the expressive aspects of a compilation remain copyrightable.

Accordingly, I conclude that the fact that the copied menu structure and first letters of the menu commands may be used to specify executable operations does not bar a finding that these elements are copyrightable. For the reasons explained in detail in Sections II.A.1 and II.B2.b of the Phase I Opinion, the structure of the menu tree including its designated keys for invoking commands (*i.e.* what Borland copied into the phantom menus) may also be viewed, in a light favorable to Borland, as a type of selection and arrangement of the executable operations in Lotus 1-2-3. The fact that the phantom menus may be used by a "reader" (whether directly or through the text of a macro) to access Lotus 1-2-3's executable operations does not compel a conclusion that this constitutes a system. *See also Paperback, 740* F.Supp. at 72-73 (rejecting argument that the macro language is a non-copyrightable element of the program).

Similarly, a program's computer code may be described as a system. Computer code consists of a series of commands that the user issues to the computer by running the program. The commands are written in such a way that the computer code as a whole is a system for communicating the program's functionality to the computer. Nevertheless, Borland cannot and does not dispute that computer code is copyrightable.

Borland's contention that the menu tree structure is a system, or is strictly utilitarian in nature, raises again the same problem of defining the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 that I have addressed since the beginning of this case. Borland wishes the court to define the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 as including the ability to interpret macros written for use with Lotus 1-2-3. I rejected this contention in deciding the parties' cross motions for summary judgment, *see* 799 F.Supp. at 212-14, 216-17, and I reject this

argument for the same reasons here. The fact that users of Lotus 1-2-3 have created macros in reliance on expressive aspects of Lotus 1-2-3 does not convert that expression into a part of the "system." That Borland wishes to copy protected expression contained in Lotus's menu tree for what Borland contends is a utilitarian purpose also does not turn that expression into a "system" under copyright law.

> *See Id* at 213-14. *See also Mazer v. Stein*, 347 U.S. 201 [,74 S.Ct. 460, 98 L.Ed. 630] (1954) (use of statue as lamp does not bar copyright protection on expression in statue); 1 Nimmer on Copyright § 2.18[D] at 2-207 ("If, however, there is a copying of the copyrightable expression, then an infringement should be found, even if the defendant employs the material for use rather than explanation.").

Borland nevertheless contends that according copyright protection to the menu structure and first letter of the command names would impermissibly protect a "discretionary pattern of events or processes." Section 102(b) provides:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b). Borland quotes a proposal by the Information Industry Association to include the following language in Section 102(b):

> However, copyright protection may exist in a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes.

(Docket No. 345 at 8 (quoting Copyright Law Revision: Hearings Before the Subcomm. on Courts, Civil Liberties, and the Admin. of Justice of the House Comm. on the Judiciary, 94th Cong., 1st Sess. 334 (1975))). Because Congress did not

adopt this "exception," Borland contends that Congress manifested an intent that a "discretionary pattern of events or processes" is not protected under copyright law and should be determined to be a "system" under section 102(b).

Nothing Borland presents to the court, however, provides any explanation of why Congress did not include this language in the statute. Failure to adopt this language may manifest a Congressional belief that the language was unnecessary or irrelevant. Indeed, Congress did not adopt any statutory language denying copyright protection for "a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes."

In any event, the explanatory language accompanying the proposal indicates that the proposal was intended to assure that copyright protection is accorded to nonliteral aspects of computer program code. Thus, Borland's argument may be interpreted as a contention that nonliteral aspects of computer code are not copyrightable, under section 102(b). If so, Borland's argument has been consistently rejected by the courts, including this one.

> *See Paperback*, 740 F.Supp. at 54; Arthur R. Miller, Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?, 106 *Harv.L.Rev.* 978, 994-96 nn. 78-88 (March 1993) (collecting cases). *See also Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir.1992) ("We have no reservation in joining the company of those courts that have already ascribed to this logic [that non-literal structures of computer programs are protected by copyright].").

I again reject the argument that copyright cannot extend to nonliteral aspects of a computer program. Moreover, Borland is wrong when it argues that, by failing to adopt the proposed statutory language, Congress manifested any intent concerning the definition of "system" in section 102(b).

Borland next argues that its phantom menus must constitute a "system," or be found "utilitarian" in nature, because the

phantom menus do not appear on the screen and are not communicated to the user.

This argument is readily dismissed. A user may never see computer code, but copyright protection is accorded to the code. Borland proposes a distinction on the ground that computer code may still be printed and read by someone. This distinction is vacuous for three reasons. First, the phantom menus may also be printed out; exhibit 517 is one copy. If one accepts the proposition that nonliteral aspects of computer code are copyrightable, the fact that the printed form of what Borland copied is not identical to any Lotus code or is not actually displayed to the user is not material to a finding that the Lotus menu structure contains copyrightable expression. Second, the fact that the phantom menus are not displayed does not mean that the user does not know they are there. Finally, copyright protection has been accorded to forms of computer code that are not generally intelligible to humans. *See Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (object code), *cert. dismissed*, 464 U.S. 1033, 104 S.Ct. 690, 79 L.Ed.2d 158 (1984).

Last, Borland argues that it has removed anything expressive in the Lotus menu tree from its phantom menus, leaving only the first letters of the command names as "markers" of the program's functionality. *See* Docket No. 311 at 25-26. This is not the case. The letters serve as "markers" of the Lotus menu tree structure. Claiming that the "markers" are part of the functionality again incorrectly assumes that the "idea" or "function" of Lotus 1-2-3 includes menu structure and first letters of the command names. Moreover, for the reasons explained in the Phase I Opinion, I conclude that the menu tree structure is original and was not dictated or even substantially limited by functional considerations at the time of its development. Accordingly, I also conclude that the menu structure and organization (including the first letter of the commands, used to mark the structure) are part of the protectable expression found in the Lotus 1-2-3 program.

In sum, I conclude that the Lotus menu structure, organization, and first letters of the command names are not within

the definition of the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program and constitute part of the protectable expression found in the program.

2. *Borland's arguments concerning earlier "holdings" of this court.*

In arguing that the copying found in Borland's phantom menus is permissible, Borland places considerable reliance on what it characterizes as earlier "holdings of this court that expressly authorize a macro translator such as the Key Reader." *See* Docket No. 345 at 1-3. As a preliminary matter, before the Key Reader complaint in this case, no party before this court had ever pressed any actual dispute or controversy between two parties over whether a specific macro "translation" facility, standing alone, would infringe Lotus's copyrights. Article III courts decide only cases and controversies actually before the court. U.S. Const. Art III, § 2. Accordingly, I could not possibly have "held" in any earlier decision that macro translation as performed by the Key Reader is permissible under copyright law.

After examining Borland's specific citations to my earlier opinions, I also observe that in making this argument Borland has done a bit of interpretative twisting of the language of my earlier opinions. In earlier opinions, I have rejected a defendant's arguments that were based on the incorrect premise that the defendant was somehow required to copy specific elements of the Lotus 1-2-3 program.

> *See Paperback*, 740 F.Supp. at 78 (no showing that copying of entire interface is necessary for macro translation); *Borland II*, 799 F.Supp. at 218 (distinguishing *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755[, 64 S.Ct. 1265, 88 L.Ed. 1584] (1944), as involving a system requiring use of only descriptive words, while in this case, varying full-word menu command names is not the only alternative for creating a different expression of the menu tree idea).

Contrary to Borland's arguments, I do not interpret the language in any of my earlier decisions rejecting these arguments as even remotely "authoriz[ing]" Borland to create its Key Reader. Explaining that a defendant's argument is overboard by use of an example is hardly equivalent to "holding" that the example is legally permissible. Moreover, in explaining my earlier decisions, I have twice observed that copying of expressive aspects of Lotus 1-2-3 may not be permissible under copyright law, even if that is the only way to achieve macro compatibility. *See Paperback,* 740 F.Supp. at 69; *Borland II,* 799 F.Supp. at 214. In *Borland II,* I stated that "I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3." 799 F.Supp. at 214. In short, I reject Borland's attempt, by selective reading and interpretation of language from earlier opinions, to twist discussions contained in those opinions into "holdings" of this court.

Finally, much of Borland's argument rests on statements concerning whether a "macro translator" *such as in Excel 2.1* infringes. Borland attempts to equate Key Reader with Excel 2.1 by labeling the Key Reader as a "translator." As explained above, however, Borland has expanded the term "translator" from the one-time translation done in Excel 2.1 to include on-the-fly interpretation as done in Borland's programs. In *Borland II,* I expressly did not decide issues concerning one-time translation, 799 F.Supp. at 214 (quoted *supra),* let alone issues concerning an expanded definition of macro "translator"—a definition including on-the-fly interpretation of macros.

D. Substantial Similarity.

For the following reasons, I find that what Borland copied into the Key Reader phantom menus is substantially similar to Lotus 1-2-3.

Contained in the phantom menus is a virtually identical copy of the menu structure and organization of Lotus 1-2-3, using the first letter of command names and other symbols to

delineate that structure. Although Borland's copy may use a different way of representing the menu structure than is contained within the Lotus code or displayed to the user while running Lotus 1-2-3, I find that what was copied by Borland (details of expression of the menu structure) is virtually identical to details of expression of the Lotus 1-2-3 program's menu structure.

I also conclude that differences in the method Borland uses to represent the menu tree structure in its phantom menus file does not negate a finding that the copied expression of the menu tree structure is substantially similar to the Lotus 1-2-3 program. First copyright law protects nonliteral aspects of a copyrighted program. Thus, one need not copy the specific code of a program to infringe copyrights in the program. Second, Borland's copying is analogous to a translation. No identity of words may exist in a translation into French of a book written in English. Nevertheless, the translated copy infringes by making a virtually identical copy of the book, but using a different method for representing the words and grammar structure. *See* 17 U.S.C. § 101 ("derivative work" includes translation); 17 U.S.C. 106(2) (copyright owner has exclusive right to prepare derivative works). Here, Borland created a virtually identical copy of the Lotus menu structure, but translated (nearly verbatim) the menu structure into a different language for representing menu structures.

Borland next contends that the fact that the phantom menus are not displayed on the screen prevents a finding of substantial similarity. I reject this contention for the same reason that I rejected Borland's argument that Borland copied only unprotected expression because the phantom menus are not displayed. *See* Section II.C.1, *supra.* That is, copyright law does not require that a program display its source or object code for copied elements of the code (literal or nonliteral) to be substantially similar.

In addition, I find that a user would understand that Borland was using a copy of the Lotus menu structure when executing a macro. Consider, for example, the "/{?}" macro described above. With this macro, the user can traverse the

Lotus menu structure precisely as though typing keys into the Lotus program. Similarly, Key Reader executes macros by sequentially parsing the (Lotus) macro text. This is demonstrated most clearly when one steps through macros in the Borland macro debugger. In addition, Borland's experts contend that a user may not even look to the full command names when using Lotus menus or macros. *E.g.* Liddle Dec'n, VI:11, ¶ 53. I accept this point and reject Borland's premise that the Borland programs do not demonstrate to the user that Borland copied Lotus's menu structure.

Finally, I find that the Lotus menu structure, including the first letters of the command names that mark that structure, constitutes a substantial part of the Lotus expression. What Borland copied into Key Reader is that portion of the Lotus program that determines precisely how the user may select from among the executable operations in the program. Qualitatively, Borland's phantom menus copy a substantial part of what a user would think of as constituting the Lotus program: the menu structure and the sequences of keystrokes for invoking each of Lotus's executable operations.

III. Affirmative Defenses.

Borland raises four affirmative defenses: waiver, laches, estoppel, and fair use. Each is considered below.

A. Waiver.

The parties agree that in order to succeed in its waiver defense, Borland must prove that Lotus voluntarily and intentionally relinquished a known right. *See* Docket No. 311 at 26; Docket No. 338 at 20. That is, Borland must prove (1) voluntary and intentional relinquishment (2) of a right that Lotus was aware that it had.

Borland points to six statements that it contends demonstrate an intention to relinquish any right to assert claims against Key Reader. The first three statements were made before the Key Reader had been introduced into any Borland

product. For these statements, Lotus clearly could not have intended to waive claims specifically against Key Reader; Lotus was unaware of Key Reader. To overcome this obstacle, Borland contends that these statements reflect the intentional relinquishment of a known right to claim that *any* kind of macro "translator" infringes.

The three statements made before Key Reader was released were made by the president of and counsel for Lotus during proceedings in the *Paperback* case. *See* Manzi Dep'n, IX:41; Lemberg Dep'n, IX:24; Trial Transcript, VIII:B.16. Each of the statements concerns whether Lotus intended to assert a claim of infringement against macro translation as performed by Excel.

Borland first cites a statement by Jim Manzi, President of Lotus, made during a deposition in the *Paperback* case. The statement ends with the conclusion that "as far as I know, we have no issue of that [Excel] product." Entirely missing from this statement is any evidence (1) that Manzi was aware of a right to prevent copying found in Excel (even this assumes, without support, that (a) there was copying, and (b) Manzi was aware of that copying), or (2) that Manzi intended to waive a claim against products other than Excel, with respect to any type of macro translation. This statement does not reflect an intention to relinquish any known rights.

Borland next cites statements by Lotus's General Counsel, Thomas Lemberg. The Lemberg statements, in evidence under seal, were not made in a context associated with waiver of rights. Accordingly, I cannot find a manifested intention to relinquish known rights based on these statements. In addition, having examined the Lemberg testimony, I also find that Mr. Lemberg was referring to one-time translation of macros into a different macro language, rather than macro translation in general, and more specifically, rather than referring to any program that executes macros by reference to copies of Lotus's menus. Whether or not the distinction between one-time translation and direct execution of macros makes a material difference under copyright law is not significant; a party may waive rights without following the specific contours of

copyright law. In any event, I find that the Lemberg statements do not demonstrate an intent to relinquish either a specific right or more generally a right to assert claims against all forms of macro translation or execution.

Finally, Borland cites the *Paperback* trial transcript at a point where Lotus's counsel, Henry Gutman, responded to a defendant's contention that the defendant "had to" copy the entire Lotus user interface into its own spreadsheet product. Gutman argued that the premise of this argument is wrong because Excel had made a commercially successful product that did not copy the full interface, and in fact, included the ability to "translate[ ] and run in Excel" Lotus macros. VIII.B.16 at 45. Based on the materials before me, I find that (1) this statement was intended to discredit a defense argument and does not demonstrate that Lotus intended to waive any right, and (2) this does not demonstrate an intent to waive claims against all types of macro translation as opposed to Lotus's understanding of Excel's one-time translation.

Moreover, each of the three statements cited above was made in the context of an infringement suit against companies that had copied the entirety of Lotus's user interface. Up to the time of the three statements cited by Borland, Lotus had never been faced with issues raised by a program that did not fully copy the Lotus 1-2-3 interface but executed Lotus macros on-the-fly. Accordingly, for each of the statements I find that Borland has not proved (1) that, at the time of the statement, Lotus was aware of a right to bar copying of its menus when the copied menus were used only to execute macros on-the-fly, or (2) that Lotus intended to relinquish its right to assert claims against a program that directly executes Lotus macros by reference to copies of the Lotus menus, rather than by one-time translation of the macros into the product's own macro language.

Borland next cites three statements made by Lotus after Key Reader had been introduced into Borland's products. The first is a response to a Borland request for admissions. According to the response, Lotus admitted that

> it does not contend that the . . . ability to execute "macros" originally written using Lotus 1-2-3 by means of a conversion or translation program, standing alone, would infringe . . . , but qualifies its admission to state that Lotus does contend that the 1-2-3 User Interface of Quattro and Quattro Pro, including, *inter alia*, its use of the 1-2-3 menu structure to provide the ability to execute macros originally written using Lotus 1-2-3, impermissibly copies protected expression contained in Lotus 1-2-3.

See VII:B.1. Later in the same document containing Lotus's response to a request for admission, Lotus objected to a Borland request for an admission that the ability to execute Lotus macros while in Quattro Pro's native mode does not infringe. Lotus's objection was based on the ground that, as far as Lotus knew, Quattro Pro had no such ability. *See* VII:A.7.

This admission does not constitute a waiver of a right to assert infringement claims against Key Reader. Lotus explicitly stated that it *does* contend that use of the Lotus menu structure to provide the ability to execute macros is impermissible, as done in Borland's emulation interface. For the reasons explained in Section II.B above, I find that the Key Reader executes macros by using copies of the Lotus menu structure in a way very similar to the way that the 1-2-3 emulation interface executed macros. The qualification to the admission demonstrates an intent to pursue claims against a program doing precisely what Key Reader does.

Borland contends, however, that the reservation should be interpreted as objecting to a facility such as Key Reader only when it is a part of an emulation interface, and not when it is incorporated into Borland's native menu interface. Considering the circumstances of the admission, however, I find that such an interpretation is unreasonable. First, under Borland's proposed interpretation, the qualification to the admission does not qualify the scope of the admission. The admission without the qualification was already limited to macro translation "standing alone." Second, when specifically asked about running Lotus macros while using the native menu interface, Lotus's answer demonstrated that Lotus was not then aware of Borland's Key Reader feature. It is, therefore,

more reasonable to interpret Lotus's response as (1) a statement that Lotus was not contending in this suit that one-time translation or conversion infringes (a point very distinct from execution of Lotus macros by reference to "phantom" menus), or (2) a statement that Lotus was not contending that translation infringes because there was no need to make that claim in this case—Borland's emulation interface infringed and the scope of that infringement was sufficient for claims against the products then at issue.

If Borland wished later to be in a position to make its present assertion of waiver, it should have made full disclosure up front about Key Reader and obtained an unambiguous response. Borland's present claim of waiver is an attempt, wholly without merit, to escape the consequences of its own litigation strategy. Lotus was entirely free to respond as it did without waiving any right. I find that Lotus's response to the request for admission does not demonstrate an intent to relinquish a known right to assert claims against Key Reader, or a feature like Key Reader.

Borland next points to a statement by Lotus's counsel at a conference in June 1991. *See* VIII:B.17. In making the statement, Lotus's counsel advised the court that if Borland "include[d] 1-2-3 menus and alternate user interface" in a product Lotus believed was about to be released, Lotus would seek a preliminary injunction. Borland's contention that this constitutes a waiver of claims against any program that does not "display" the menus approaches, if not crosses, the line into frivolousness.

Finally, Borland points to a footnote in one of Lotus's briefs. *See* VIII:B.6. Here, Lotus was responding to Borland's contention that it *had* to copy and display the entire menu tree to achieve macro compatibility. In the footnote, Lotus argues that Borland was ignoring *this court's* distinction in *Paperback* between display of menus and executing macros by "interpretation, translation or conversion (as then existed in Excel or the foreign language versions of Lotus 1-2-3)." The foreign language versions of Lotus 1-2-3 perform a one-time translation or conversion of macros from English releases of

1-2-3 into a macro using the foreign language version of 1-2-3. *See* Morgan Dec'n, VI:12. This does not constitute waiver for several independent reasons.

First, the language focuses on the court's distinction, not Lotus's.

Second, Lotus sought to rebut Borland's argument by pointing out that macro compatibility might be achieved without display of the menus and cited examples of one-time macro conversion but not on-the-fly macro interpretation. In the circumstances here, I do not find that impeachment of a defendant's argument by example demonstrates an intent to relinquish any rights.

Third, at the time the argument was made, Borland had not released a product containing Key Reader but without the full emulation interface. Thus, Lotus was not faced with any decision over whether to assert claims against a program that copied Lotus's menus, but only for on-the-fly interpretation. I find that this statement does not demonstrate an intention to relinquish a known right to pursue infringement claims against a program containing Key Reader.

In sum, considering the statements proffered by Borland both separately and as a whole, I find that Borland has not proved that Lotus has waived a right to pursue its claims against Key Reader.

B. Laches.

To succeed in its laches defense, Borland must prove that (1) Lotus inexcusably or unreasonably delayed in raising claims based on Key Reader, and (2) Borland was prejudiced by this delay.

1. Delay.

Lotus first raised claims of infringement based exclusively on Key Reader in its motion for leave to file supplemental complaint, filed December 7, 1992. (Docket No. 250.) Borland contends that this is more than two years after Key Reader was introduced. As explained above, however, Key

Reader was introduced into products that also contained the emulation interface (Quattro Pro 2.0, 3.0, and 4.0).

I begin by examining whether Lotus's delay was unreasonable or inexcusable during the time period when Borland's products included both the emulation interface and Key Reader. I then examine the delay in filing suit after Borland removed the emulation interface and marketed a product containing only Key Reader, *i.e.*, after Quattro Pro 4.01 was announced in August 1992.

For the following reasons, I find that the delay in filing suit before Borland introduced Quattro Pro 4.01 into the market was both reasonable and excusable.

First, even assuming Lotus was aware or should have been aware of Key Reader in Quattro Pro 2.0, 3.0, and 4.0, a fact disputed by Lotus. there is little reason that Lotus should have known (and no evidence that Lotus did know) that Key Reader was anything other than a link to the macro execution facility that is contained in the 1-2-3 emulation interface. In these circumstances, it would have been reasonable for Lotus to believe that if the emulation interface were removed Borland's Key Reader would not function. Indeed, Borland's Chief Executive Officer stated at a deposition in May 1991 that Quattro Pro would not be macro compatible with Lotus 1-2-3 if the 1-2-3 compatible menu tree were removed. *See* I:13 at 52. Thus, claims against the emulation interface might then have appeared to protect Lotus fully, without the added cost of asserting separate claims against Key Reader.

Second, before Quattro Pro 4.01 was released, Lotus was pursuing its strongest claims against all of Borland's existing spreadsheet products, *i.e.*, claims based on the emulation interface. In the circumstances of this case, I cannot say that Lotus was required to assert claims against each feature that Borland added to its (already) infringing products or be faced with the prospect of losing the right ever to assert claims against a program containing that feature. Until the emulation interface was removed, Lotus was not faced with a product that would not be found infringing without a determination that Key Reader infringes. I conclude that it was reasonable

to wait to assert claims against Key Reader until resolution of this dispute was necessary and could be raised explicitly as a live issue.

Third, to the extent Lotus was expressly pursuing claims against macro execution as performed by Borland's emulation interface, *i.e.*, on-the-fly macro interpretation by reference to Lotus's menu structure (*see* VII:B.1, quoted *supra*), Lotus was pursuing claims against something very similar to Key Reader.

Considering these factors, I find that Lotus's failure to bring an action specifically directed at Key Reader for the time period from the incorporation of Key Reader in Quattro Pro 2.0 until release of Quattro Pro 4.01 was both reasonable and excusable because of Lotus's pursuit of infringement claims against the emulation interface in each of Borland's then existing products.

Next, I consider whether Lotus's delay in filing suit after Quattro Pro 4.01 was released was unreasonable or inexcusable. Quattro Pro 4.01 was released after the *Borland II* decision was issued on July 31, 1992. Lotus filed its motion for leave to file supplemental complaint approximately four months after Quattro Pro 4.01 was made public. A few months before Quattro Pro. 4.01 was released, Lotus gave notice of an intent to preserve claims against Key Reader (in April 1992, *see* VIII:A.5). Borland acknowledged this notice. (*See* VIII:A.2.) I also note the propriety of taking some time to investigate how Key Reader works before filing a complaint. In light of the brevity of the time period, the nature of investigation of Key Reader that was reasonably required, and Lotus's earlier notice of its intent to preserve claims against Key Reader, I find that Lotus did not unreasonably or inexcusably delay in bringing an action specifically directed at Key Reader.

Further, I reject Borland's contention that the motion to supplement the complaint was filed so late in the proceedings on the original complaint that Lotus's delay was unreasonable. It was the late (and secretive) timing of the release of Quattro Pro 4.01 that resulted in the late timing of Lotus's

claims. Moreover, Lotus agreed to a short time schedule for preparing and presenting its claims for both of the first two Phases of the trial. Accordingly, I find that Lotus's delay in filing claims against Key Reader was reasonable and excusable, despite the advanced stage of proceedings under the original complaint.

2. Prejudice.

As an example of alleged prejudice caused by Lotus's delay in bringing suit, Borland contends that an appeal of copyrightability issues to the First Circuit has been delayed. To make this claim, Borland contends that it removed the emulation interface (but not the Key Reader) in order to expedite an appeal to the First Circuit. The record is barren of evidence to support what counsel contends was Borland's motive for removing the emulation interface.

In addition, even assuming *arguendo* that this unsupported assertion is true, I find that Borland has not proved prejudice. For the reasons explained on the record at the September 23, 1992 conference, I denied Borland's motion for interlocutory appeal. This was before Lotus moved for leave to include claims against Key Reader in this case. Thus, before Lotus filed its motion for leave to supplement, I had already concluded that an interlocutory appeal was not appropriate. Borland has not demonstrated that it could have secured an interlocutory appeal of the summary judgment decision, even if Lotus never asserted claims against Key Reader.

To demonstrate prejudice, Borland also points to costs in developing and advertising Key Reader. Even assuming these costs were appreciable, a finding that Borland provides little evidence to support, I find that these costs were not incurred *as a result* of any delay by Lotus. Of course, Lotus had not delayed at all in bringing suit before Key Reader was introduced into a Borland product. In any event, as to Quattro Pro 2.0, 3.0, and 4.0, these products have been found to infringe whether or not Key Reader was present. Borland points to no costs with respect to versions 2.0, 3.0, and 4.0 that might have been avoided had Lotus asserted infringement claims directed

at Key Reader earlier. Moreover, Borland did not remove the emulation interface from Quattro Pro 2.0, 3.0, or 4.0, all released after Lotus had filed the complaint in this action. I find that Borland would have included Key Reader along with the emulation interface in its products whether or not Lotus immediately asserted that Key Reader infringes.

With respect to Quattro Pro 4.01 and later releases, Borland released these programs months after Lotus had sent a message (that Borland had acknowledged) that Lotus was seeking to preserve claims against programs containing Key Reader. (*See* discussion in preceding section.) I conclude that any prejudice suffered by Borland's inclusion of Key Reader in Quattro Pro 4.01 and later releases was not caused by delay on Lotus's part.

C. Estoppel.

As was true with respect to the estoppel claims in Phase I, the parties apparently agree that to establish a defense of estoppel in relation to Phase II issues, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. As I did in the Phase I Opinion, again without endorsing this specific formulation of the standard for deciding a claim of estoppel, I proceed, to Borland's advantage, to apply the standard stated by the parties.

In determining whether Borland has proved reasonable reliance on any conduct by Lotus, I consider first Borland's inclusion of Key Reader in Quattro Pro versions 2.0, 3.0, and 4.0, then turn to inclusion of Key Reader in Quattro Pro 4.01 and later releases.

Versions 2.0, 3.0, and 4.0 were released after Lotus filed suit claiming that Quattro and Quattro Pro infringe. Nevertheless, Borland included the emulation interface in each of these products. Accordingly I find that, even if Lotus had clearly and unequivocally stated an intention to assert claims against something like Key Reader, Borland would have

included Key Reader in these products anyway. Borland did not rely on any statement by Lotus in developing Key Reader.

For the following reasons, I also find that Borland has not proved that a reasonable person would have relied on any conduct by Lotus in developing or advertising Key Reader in versions 2.0, 3.0, and 4.0.

Borland contends first that it relied on Lotus's original complaint, ¶¶ 17-19, to conclude that Lotus would never assert claims against a program that copied Lotus's menu structure within the computer code but did not display the Lotus menu structure on the screen. This contention is without merit. The complaint alleged claims against products that did not have a Key Reader. The complaint focused on the emulation "user interface" because that is precisely the portion of the programs (in existence at the time the complaint was drafted), both as displayed on the screen and as implemented by the computer code, that infringes. In any event, without Borland's computer code, there is no Borland emulation interface. The complaint cannot be read in any way other than as asserting claims for infringement based on Borland's computer code. Finally, the premise to Borland's argument, that Lotus knew or should have known enough about the computer coding of Quattro and Quattro Pro to make specific allegations in the complaint about the computer code is without support in the record. Borland's claim that it reasonably relied on the original complaint as demonstrating an intention by Lotus to pursue infringement claims based only on Borland's display of the interface and not on Borland's code is baseless.

Borland next cites three statements, made after Key Reader was first released in Quattro Pro 2.0, by Lotus executives or counsel. These three statements are the same statements that Borland cited in its waiver argument, *i e.*, (1) the admission, (2) the statement at the June 1991 conference, and (3) a footnote in one of Lotus's briefs. For the reasons explained in Section III.A, *supra* I conclude that Borland could not reasonably rely on any of these statements as demonstrating an intent not to pursue claims against Key Reader. Moreover, to

the extent that the statements and the definition of macro "translation" are ambiguous, reliance on those statements would be unreasonable. A reasonable person or company in Borland's position would have sought clarification before relying on the statements as a position adopted by Lotus.

In addition, each of these statements was made after Key Reader was introduced into Quattro Pro 2.0. Borland, when originally developing and introducing Key Reader, could not have relied on statements made later, after Key Reader had been developed. Nevertheless, Borland contends that it relied on these statements in retaining Key Reader in Quattro Pro 4.01 and later releases. As explained above, however, in April 1992, Lotus clearly stated (and Borland acknowledged) that Lotus wished to preserve claims against Key Reader. In light of this, I find that for Quattro Pro 4.01 and later releases, Borland could not have reasonably relied on the cited statements as demonstrating an intent by Lotus not to assert claims against Key Reader.

Finally, Borland's General Counsel avers that, when recommending to keep Key Reader in the program, he relied "most directly" on the court's language in the *Borland II* summary judgement opinion. *See* VI:9, ¶¶ 13-14. First, for the reasons explained in Section II.C.2, I find that Borland's selective reading of the opinion and twisted interpretation of the court's language is unreasonable; reliance on this language, therefore, also would be unreasonable. Second, I am not aware of any precedent for basing an estoppel defense not on conduct of the party alleged to be estopped, but on conduct of the court. Borland offers no authority to support this proposition. Instead, Borland contends that the court's language reflects Lotus's arguments. I reject this contention. To support such a contention, Borland would have to place in evidence Lotus's arguments and conduct, rather than simply pointing to the court's decision. I have examined the language Borland points to in Lotus's earlier briefs and argument and reject Borland's contention that Borland reasonably relied on Lotus's statements or conduct.

In any event, to the extent Borland relied on the court's language in deciding to include Key Reader in Quattro Pro 4.01, Borland was not relying on Lotus's conduct.

Having considered the several instances of conduct cited by Borland both individually and as a whole, including any delay in filing suit, I find that Borland has not proved its estoppel defense.

D. Fair Use.

Borland contends that copying of the Lotus menu structure and first letters of the menu commands is a fair use of Lotus's copyrighted programs. The fair use doctrine is an "equitable rule of reason." *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 448, 104 S.Ct. 774, 792, 78 L.Ed.2d 574 (1984) (quoting legislative history). The doctrine

> permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster.

Stewart v. Abend, 495 U.S. 207, 236, 110 S.Ct. 1750, 1768, 109 L.Ed.2d 184 (1990) (quoting *Iowa State University Research Found. Inc. v. American Broadcasting Cos.*, 621 F.2d 57, 60 (2d Cir.1980)).

Congress set forth four statutory factors to guide consideration of what constitutes a fair use. Section 107 of the Copyright Act provides that:

> the fair use of a copyrighted work . . . for purposes such as criticism, comment, news reporting, teaching . . . , scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—
>
> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
>
> (2) the nature of the copyrighted work;

> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
>
> (4) the effect of the use upon the potential market for or value of the copyrighted work.

17 U.S.C. § 107. These factors are not exclusive. Each case is to be decided on its own facts. *See Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 560, 105 S.Ct. 2218, 2230, 85 L.Ed.2d 588 (1985).

1. "[P]urpose and character of the use."

The first statutory factor focuses primarily on whether the copied materials are used to secure profit.

> The crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price.

Harper & Row, 471 U.S. at 562, 105 S.Ct. at 2231. Section 107(1) contrasts a use that is "of a commercial nature" with a use that is for "nonprofit educational purposes." According to the preamble in section 107, copying for "criticism, comment, news reporting, teaching . . . , scholarship, or research may be closer to being a "nonprofit educational use" than a commercial use.

In the instant case, there can be little dispute that Borland copied from Lotus for "commercial" purposes. Borland sells its spreadsheet products for profit. The Key Reader is included to make the program more attractive to potential customers than have Lotus 1-2-3 macros. Borland has copied Lotus's menu structure in order to enhance profits from sales of its spreadsheet products, without having secured a license to do so.

Borland tries to minimize the impact of having a profit motive on its fair use defense by contending that Key Reader is actually used by consumers and only for interpreting the macros that the consumers have written. (Docket 345 at 19.) The consumer's use of the part of Quattro Pro that Borland

copied from Lotus, however, does not change the character of *Borland's* use of the copied materials—to please consumers and increase sales. To contend that copying is closer to a fair use because customers want the copied materials is entirely without merit. "It is fundamentally at odds with the scheme of copyright to accord lesser rights in those works that are of the greatest importance to the public." *Harper & Row*, 471 U.S. at 559, 105 S.Ct. at 2230.

Borland nevertheless contends that the present case is "almost identical" to *Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.*, 964 F.2d 965 (9th Cir.1992), *cert. denied*, ___ U.S. ___, 113 S.Ct. 1582, 123 L.Ed.2d 149 (1993). In *Galoob*, the alleged infringer sold a product (the "Game Genie") that allowed private users of Nintendo's copyrighted games to use Nintendo's games in an enhanced fashion. Nintendo did not argue that the Game Genie directly infringes Nintendo's copyright. Rather, Nintendo argued that *use* of the Game Genie with a Nintendo game creates a derivative work of the copyrighted Nintendo game. *See id.* at 970. Thus, Nintendo's infringement claims concerned only whether Galoob *contributed* to a consumer's direct infringement of Nintendo's copyright when the consumer used the Game Genie with a Nintendo game at home. For this reason, the opinion concerning fair use explicitly restricted its focus to whether a consumer's private use of the Game Genie was a fair use (and not on Galoob's use of the Game Genie—sale to the consumer for profit). *Id.* Private use by a consumer is a nonprofit use rather than a commercial use. *See id. See also Sony Corp.*, 464 U.S. at 449, 104 S.Ct. at 792.

The present case is readily distinguished from *Galoob*. Lotus contends and I have found that Borland copied and that the Quattro Pro products, including Key Reader, *directly* infringe Lotus's copyright. The fair use issues in this case properly focus on Borland's use (and not the consumer's use) of the infringing products. Otherwise, a book store could copy and sell "Gone With the Wind" without permission. The book store would rely on the consumer's private use of the copied material to claim that its copying is a fair use.

Borland cites *Sega Enters. Ltd v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir.1992) for the proposition that "the commercial nature of a use is a matter of degree, not an absolute." In finding that the commercial nature of the defendant's use in *Sega* was of a lesser degree, the Ninth Circuit relied on the fact that the copied materials were not included in defendant's final product. *Id.* Here, however, what Borland copied from Lotus is included in the product it sells to the public.

Borland also relies on *Sega* to argue that Borland has performed its own creative work in developing Quattro Pro, and therefore, Key Reader merely "supplements" Borland's own creative work. Assuming this to be true does not change the character of Borland's commercial use of the copied material. Most of Borland's own creative work, including its own menu structure, exists in Borland's programs independently of whether or not Borland includes a Key Reader in its product. Borland used the copied materials to please customers and increase sales, independently of the creative work it did. In addition, the *Sega* case is readily distinguished. *Sega* involved "intermediate" copying; the defendant did not include copied materials in the final product. 977 F.2d at 1522. Here, Borland "supplemented" the product it sells to consumers with a facility including the impermissibly copied materials.

In sum, I find that the "purpose and character" of Borland's use of Key Reader is entirely commercial. This conclusion concerns just a single factor among a number of factors a court weighs in deciding a fair use defense. Nevertheless, when a defendant's use of copied materials is determined to be commercial, that use is "presumptively an unfair exploitation of the monopoly privilege that belongs to the owner of the copyright." *Abend*, 495 U.S. at 237, 110 S.Ct. at 1768 (quoting *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793). *See also Harper & Row*, 471 U.S. at 562, 105 S.Ct. at 2231.

2. "[T]he nature of the copyrighted work."

This statutory factor focuses on locating a work, according to its nature, on the spectrum from factual works to fiction or fantasy. "The law generally recognizes a greater need to

disseminate factual works than works of fiction or fantasy." *Harper & Row*, 471 U.S. at 563, 105 S.Ct. at 2232. Similarly, a work largely dictated by functional considerations is closer to the "factual works" end of the spectrum than the fiction end. *See Sega*, 977 F.2d at 1524.

Also, even for factual works,

> there are gradations as to the relative proportion of fact and fantasy. One may move from sparsely embellished maps and directories to elegantly written biography. The extent to which one must permit expressive language to be copied, in order to assure dissemination of the underlying facts, will thus vary from case to case.

Harper & Row, 471 U.S. at 563, 105 S.Ct. at 2232 (quoting Gorman, *Fact or Fancy? The Implications for Copyright*, 29 J. Copyright Soc. 560, 561 (1982)). Thus, weighing this factor requires a determination of how much freedom of expression the author had in formulating the copyrighted expression. If there was little freedom, the expression is like a bare map. With greater expression found in the work, the scope of copyright protection is greater and a finding of fair use less likely.

For the reasons explained in the Phase I Opinion, I concluded that Lotus had substantial freedom of expression in forming its menu tree. The degree of freedom of expression in creating a menu structure using only first letters of command names is no less than the degree of freedom in designing Lotus's menu tree using full command names. Nevertheless, the menu tree is based on the set of executable operations selected for Lotus 1-2-3. Accordingly, I find that the copyrighted work in this case is similar to a "factual work," but that copying from the substantial expression found in the menu tree is not remotely necessary for disseminating the underlying executable operations (or "facts"). I find that this factor does not weigh significantly in either party's favor.

Borland contends that the "utilitarian nature" of the menu tree weighs in favor of a finding of fair use. Lotus's copyright does not, however, extend to aspects of the program that are utilitarian or functional and not expressive. Moreover, Borland's reliance on *Sega* is misplaced. The *Sega* court determined that copying of protected expression was necessary in order to gain access to unprotected aspects of the program. For this reason, the *Sega* court determined that copyright protection for the code in *Sega* was entitled to a lower degree of protection than other literary works (*i.e.*, less protection against the "intermediate" copying done by the defendant in that case). 977 F.2d at 1526. Here, Borland's copying was not necessary to examine unprotected aspects of Lotus 1-2-3.

3. *"[A]mount and substantiality of the portion used in relation to the copyrighted work as a whole."*

Borland created a virtually identical copy of the menu structure and first letters of command names from Lotus 1-2-3's menu tree. This necessarily includes copying of the keystroke sequences used by Lotus to access executable commands and Lotus's macro language. For the reasons explained in Section II.D, *supra*, I find that this is a substantial part of the copyrighted expression in Lotus 1-2-3.

Borland argues that "it is difficult to see how Borland could use even less of 1-2-3." Docket No. 345 at 21-22. The answer to this is simple. Borland could have copied none of Lotus's expression, or (though I do not comment on the permissibility of doing so) less than the complete 1-2-3 menu structure and first letters of command names.

The fact that Borland uses the menu tree structure and first letters of the command names only for macro translation is not significant in evaluating this statutory factor. "[A] taking may not be excused merely because it is insubstantial with respect to the *infringing* work." *Harper & Row*, 471 U.S at 565,105 S.Ct. at 2233 (original emphasis).

I find that this statutory factor weighs in favor of Lotus.

4. *"[T]he effect of the use upon the potential market for or value of the copyrighted work."*

The fourth statutory factor focuses on whether "*some* meaningful likelihood of future harm exists." *Sony Corp.*, 464 U.S. at 451,104 S.Ct. at 793 (original emphasis). To negate a claim

of fair use, a copyright holder may show either "that the particular use is harmful, or that if it should become widespread, it would adversely affect the potential market for the copyrighted work." *Id.* (holding that, for a noncommercial use, copyright holder must make this showing). "This inquiry must take account not only of harm to the original but also of harm to the market for derivative works." *Harper & Row*, 471 U.S. at 568, 105 S.Ct. at 2234.

As noted in Section III.D.1, *supra*, a commercial use of the copied material is presumptively unfair. Thus, "[i]f the intended use is for commercial gain, [the] likelihood [of future harm] may be presumed." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793.

In the present case, I find that there is a meaningful likelihood that Borland's copying negatively affects the market for and value of Lotus's copyright. Quattro Pro's functionality is largely equivalent to Lotus 1-2-3's functionality. The operations the programs perform on spreadsheets are similar in nature. From this, I infer that many users will not go to the expense of purchasing, maintaining, and updating both the Quattro Pro and Lotus lines of spreadsheet products.

Borland's inclusion of Key Reader caters to users who already own Lotus 1-2-3; *i.e.*, users that have developed macros for use with Lotus 1-2-3. This may directly affect sales of Lotus 1-2-3 programs. For example, a company using Lotus 1-2-3 may decide to shift to use of Quattro Pro rather than purchasing additional copies of Lotus 1-2-3 for new employees. Inclusion of Key Reader may also have an impact on sales of updates to or new releases of Lotus 1-2-3. For example, Key Reader increases the likelihood that a user will shift to Quattro Pro rather than purchase an update to Lotus 1-2-3.

For these reasons, I infer that there is a meaningful likelihood of harm to Lotus (by a negative impact on the market and a decrease in value of Lotus's copyright) stemming from Borland's use of Key Reader.

Borland seeks to rebut this inference with evidence that Lotus's share of the spreadsheet market has remained steady

from the time before Borland introduced Key Reader to the present. I reject this argument for the following reasons.

First, this statutory factor includes an examination of what would happen "if [the use] should become widespread." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793. This is a necessary consideration. Otherwise, a local bookstore would be free under the fair use doctrine to copy and sell a popular book (without license) because the impact on the national market is negligible. Here, if consumers widely purchase Quattro Pro intending to use Key Reader, I find that the market for Lotus 1-2-3 products and value of the Lotus copyright would be diminished.

Second, Borland provides no evidence that maintaining market share is equivalent to having a copyright of undiminished value. A number of factors other than a complete lack of impact on the market for Lotus 1-2-3 may explain a constant market share. For example, pricing of both 1-2-3 and competing products (including cut prices or lack of increase in prices of 1-2-3) may maintain a constant market share. Also, advertising might be increased to maintain market share. The evidence presented by Borland falls far short of supporting an inference that Lotus market has not been adversely affected.

Third, even if Lotus can maintain a constant market share without increased costs associated with lower prices or higher expenses, this does not demonstrate that the copying found in Key Reader has not diminished Lotus's market share from what it would be if Borland had not copied. Without Borland's copying, Lotus's market share may have increased.

In short, I find that Borland's limited evidence concerning market share is entirely insufficient to demonstrate that Lotus has not been harmed or to rebut the presumption that Borland's commercial use of Key Reader presents a meaningful likelihood that Lotus will be harmed. This statutory factor weighs in favor of Lotus.

5. Other factors.

The statutory list of factors a court may consider in deciding the merits of a fair use defense is not exclusive. As an additional factor, Borland contends that public policy dictates that Borland's use of Key Reader be considered a fair use. Borland argues that permitting Borland to market an "improved" spreadsheet product while maintaining macro compatibility for users of Lotus 1-2 3 stimulates (Borland's) artistic creativity. In essence, Borland contends that, when forming new artistic expressions, Borland should be permitted to tap into the market for existing expression by copying not just the idea but by copying from the expression of the idea. I reject Borland's arguments for the following reasons.

First, this "new" factor is just another argument concerning the "purpose and character" of Borland's use of the copied expression. Thus, I reject Borland's argument for the same reasons that I concluded that Borland's use of the copied materials is commercial. Borland is not entitled to rely on the consumers private use of the program for execution of macros to claim fair use.

Second, even when barred from copying expression, Borland was and is free to create new expressions of the ideas contained in Lotus 1-2-3. Borland presents no evidence that providing users with macro compatibility, or the ability to execute macros by reference to copies of Lotus's menus, is necessary to permit copying of the ideas found in Lotus 1-2-3. *Compare with Sega*, discussed *supra*. Borland has not demonstrated that allowing it to copy Lotus's expression would stimulate creativity more than the general scheme of providing copyright protection for an author's expression.

6. Summary.

Each of the factors to be weighed in deciding whether Borland's copying constitutes a fair use either weighs against a finding of fair use, or is entitled to little or no weight. Considering all factors, I find that Borland's copying is not a fair use of Lotus's copyrighted expression.

IV. Summary.

For the reasons explained above, I find that Borland's Key Reader infringes Lotus's copyrights. Further, I find that (1) Lotus has not waived claims against Key Reader, 2) Lotus is not barred by laches or estoppel from asserting claims against Key Reader, and (3) including expression copied from Lotus 1-2-3 in Borland's Key Reader facility is not a fair use of the protected expression.

Permanent Injunction

For the reasons stated in the July 31, 1992 Memorandum and Order, and for the reasons stated and on the Findings and Conclusions recited in the Opinion of June 30, 1993, as amended August 19, 1993 (the "Phase I Opinion"), and the Opinion of August 12, 1993, as amended August 19, 1993 (the "Phase II Opinion"), it is ORDERED, ADJUDGED, AND DECREED:

1. Defendant Borland International, Incorporated ("Borland") has infringed plaintiff's copyrights in the computer software program "Lotus 1-2-3" by its development, manufacture and sale of: (1) Quattro and Quattro Pro version 1.0, by Borland's inclusion of its "1-2-3 emulation" interface; (2) Quattro Pro versions 2.0, 3.0, and 4.0, by Borland's inclusion of its "1-2-3 emulation" interface and its "Key Reader" facility: and (3) Quattro Pro SE, 4.01 and Quattro Pro for Windows by Borland's inclusion of its "Key Reader" facility.

2. Acting under 17 U.S.C. § 502(a), the court orders that Borland, its officers, agents, servants, employees and attorneys, and all persons in active concert or participation with them, are hereby permanently enjoined from manufacturing, selling, distributing, licensing, or continuing to license for manufacture, sale, distribution or sub-licensing

(i) Quattro, Quattro Pro versions 1.0, 2.0, 3.0, 4.0, SE, 4.01, and all versions of Quattro Pro for Windows that

have been or are on the market on this date (August 19, 1993) ("Quattro Pro for Windows"), and

(ii) any work that contains in any portion, component or module thereof, a copy of the Lotus 1-2-3 menu commands and/or menu structure, in any form. Borland may at any time, however, apply to this court for modification of this clause upon a showing of good cause for determining that Borland has developed or proposes to develop a product that is within this clause but for special reasons

(a) the product is not infringing, or

(b) an alternative form of remedy allowing marketing on specified conditions, including payment of royalties, is more appropriate than an injunction.

3. In addition, upon entry of this Permanent Injunction, Borland shall forthwith take all necessary steps to terminate any existing licenses, distribution agreements or other arrangements pursuant to which it has manufactured, sold, distributed or licensed any of Quattro, Quattro Pro version 1.0, 2.0, 3.0, 4.0, SE, 4.01 or Quattro Pro for Windows and it shall, at its own expense, on or before August 27, 1993, notify each and every other party to any such license, distribution agreement or other arrangement of the existence and terms of this Permanent Injunction. Borland shall take all reasonable measures necessary to ensure, so far as it can control, that all such manufacturers, distributors and resellers comply with the terms hereof, including reasonable measures to prevent the selling of infringing products, which may include, but does not require, repurchase of products, and which does not include a general recall.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K
June 30, 1993
As Amended Aug. 19, 1993

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

OPINION

KEETON, *District Judge*.

On July 31, 1992, the court allowed, in part, a motion for summary judgment filed by plaintiff Lotus Development Corporation ("Lotus") and denied the cross motion for summary judgment by defendant Borland International, Incorporated ("Borland"). After extended procedural maneuvering, the parties agreed to try remaining liability issues without a jury. An explanation of the proceedings leading up to the trial is essential to precise identification of the issues raised by the parties in the nonjury trial of February 1-3 and March 31-April 2, 1993.

I. Earlier Proceedings.

A. Partial Summary Judgment.

This Opinion assumes the reader's familiarity with the Memorandum and Order allowing, in part, Lotus's motion for summary judgment. That document was published as *Lotus Dev. Cop. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) (July 31 Memorandum and Order). In addition, the terminology used in this Opinion follows the terminology set forth in detail in the earlier Memorandum and Order. *Id.* at 206-208. Background information appears in two earlier documents issued by this court. The first is a Memorandum and Order in this case, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass.1992). The second is an opinion in a related case involving claims of infringement of copyrights for the Lotus 1-2-3 program. *Lotus Dev. Corp v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990).

The July 31 Memorandum and Order explained the standard to be applied in this case for determining copyrightability issues:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

> that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.
>
> * * *
>
> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.
>
> THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

Borland, 799 F.Supp. at 211 (quoting *Borland*, 788 F.Supp. at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphases omitted)).

Applying this test, I determined that the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program is a menu-driven electronic spreadsheet whose

> user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are

> accessible through the paths of the menu command hierarchy.

Id at 216-17. (The meanings of menu command, menu structure and menu tree are explained at greater length below.) I concluded also that

> as a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

Id at 223 (original emphasis).

Nevertheless, I concluded that determining the scope of relief in this case depends on resolution of disputed factual contentions because Lotus contended and Borland disputed "that the copying of separable expressive elements of the Lotus 1-2-3 user interface into the Quattro programs was greater than the minimum essential to constituting a substantial part of the Lotus 1-2-3 work," which I had determined on motion for summary judgment not to be genuinely in dispute. *Id*. In other words, there is no genuine dispute of fact that the Quattro and Quattro Pro programs infringe, but fact issues remain as to the scope of impermissible copying. Specific fact issues apparent on the record at that time concerned (1) whether Borland copied the long prompts of Lotus 1-2-3, (2) whether the long prompts contain expressive elements, and (3) the extent (if any) that functional constraints limit the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See* Order Regulating Jury Trial, September 30, 1992 (Docket No. 232) at 20.

In addition, I concluded that Lotus was entitled to summary judgment against Borland on the affirmative defense of

waiver, but not on the affirmative defenses of laches and estoppel. *See Borland*, 799 F.Supp. at 222-23.

B. Further Proceedings Before Trial.

Up to the time of the court's ruling of July 31, 1992, the parties' contentions concerned issues raised in the allegations of the "original complaint" filed July 2, 1990, concerning infringement of Lotus 1-2-3 by Quattro and Quattro Pro's "emulation interface." Trial for the remaining liability issues in the original complaint was scheduled before a jury. In January 1993, this court permitted Lotus to file a supplemental complaint alleging copyright infringement by Borland in programs containing a "Key Reader" feature (which is described in some detail in the August 12, 1993 Opinion that addresses the issues of the "Key Reader" phase of the trial)—specifically in Quattro Pro versions 2.0, 3.0, 4.0, SE, 4.01 and Quattro Pro for Windows. *See* Docket 250, Exh. A.

After these developments and as the parties were preparing for trial of issues raised in the original complaint, the parties entered into a series of stipulations that altered the nature of the proceedings. *See* Stipulation and Order Regulating Trial (Docket No. 330); Stipulation and Order Regulating Key Reader Trial (Docket No. 349).

The first set of stipulations concerned trial of issues raised in the original complaint. These stipulations

> govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial").

Docket No. 330, ¶ 1. With respect to issues raised in the original complaint, the parties waived jury trial for the liability issues that had previously been scheduled for the Phase I trial.

With respect to the long prompts, the parties stipulated that:

> 1. The order of display of the long prompts within the "1-2-3-compatible" modes of Quattro and Quattro Pro

follows the order of display of the menu commands within those modes, and each such long prompt provides a short textual description of the command to which it relates.

2. Lotus shall not contend, in this action or any appeal therefrom, that Borland has copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

3. Borland shall not contend, in this action or any appeal therefrom, that Borland has not copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

4. Neither party shall contend, in this action or any appeal therefrom, that the issue of whether or not Borland copied the long prompts of Lotus 1-2-3 in either Quattro or Quattro Pro is material to any other issue that has been or will be resolved in this case.

Docket No. 330, Exh. A, ¶¶ 1-4.

The second set of stipulations "govern[s] the trial of all liability issues (including any defenses thereto) raised by Lotus' Supplemental Complaint concerning the 'Key Reader' feature (the 'Phase II Trial')." Docket No. 349, ¶ 1. For trial of the liability issues raised by the supplemental complaint, the parties waived their rights to trial by jury. *Id.*

C. Summary of Issues Before the Court.

Phase I of the trial was held on February 1-3, 1993. At that time, the issues before the court were the scope of infringement by Borland and Borland's affirmative defenses of laches and estoppel (the affirmative defense of waiver having been resolved at summary judgment). After the close of Borland's evidence, however, Borland was allowed leave to amend its answer to assert an affirmative defense of fair use. *See* Memorandum and Order, March 30, 1993 (Docket No. 353). In response, Lotus moved for judgment on partial findings. *See* Fed.R.Civ.P. 52(c). After hearing in open court and for the reasons stated on the record, I allowed Lotus's motion for

judgment on the issue of Borland's fair use defense to the original complaint (Phase I).

Phase II of the liability trial was held on March 31-April 2, 1993. The issues presented to the court included the full range of liability determinations for the "Key Reader" supplemental complaint. In addition, the parties tried Borland's defenses (to the "Key Reader" complaint) of waiver, laches, estoppel, and fair use.

This opinion addresses the issues raised in Phase I of the trial. On August 12, 1993, the court released an Opinion resolving the issues raised in Phase II.

II. Scope of Infringement in Phase I Trial.

As I understand the parties' stipulations and arguments, Lotus does not now contend that Borland copied the entire 1-2-3 interface. Rather, Lotus claims that Borland has illegally copied the Lotus 1-2-3 "menu commands" and "menu structure." Accordingly, the only issues before the court concern copying of the menu commands and structure.

Borland contends that copying of menu commands and structure is permissible because of functional constraints on formulation of the menu commands and structure. To the extent that Borland contends that the menu commands and structure as a whole are not copyrightable, Borland's contention was rejected as a matter of law at summary judgment. Nothing Borland presented at the Phase I trial alters my view that there is no genuine dispute that "a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations." 799 F.Supp. at 218.

Although I determined that no genuine dispute of fact had to be resolved in order to determine that the menu commands and structure contain protectable expression, I also determined that disputed factual contentions might have to be resolved to determine the *scope* of infringement. Among factors bearing on the scope of infringement are (1) the scope of copying, and (2) the nature of the copied work. Given the

implications of the idea and the functional considerations, what is the extent of the expression? If there is "essentially only one way to express an idea," complete copying is permissible. *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir.1988). If there are "only a limited number of ways of expressing the idea," then proof of "near identity" of copied expression is required to show infringement. *Id.* For a work "embodying only one of an infinite variety of ways of expressing an idea," duplication or near identity is not required. *Id.* at 607. Consequently, there is a "sliding scale" that determines the scope of copyright protection. Accordingly, I first examine more closely the scope of copying and then examine how much of this copying was impermissible.

A. Menu Commands and Structure—Scope of Copying.

1. Defining and describing "menu commands and menu structure."

Omitting details not relevant for resolution of any dispute in this case, one may describe the idea of menu commands and menu structure in the following way.

Each spreadsheet program described in the evidence before the court has a set of basic executable operations that a user may invoke. Each executable operation does something with data in a spreadsheet (*e.g.*, erases data in a spreadsheet cell), fixes parameters of a spreadsheet (*e.g.*, width of a column in a given spreadsheet), fixes parameters of the program (*e.g.*, hardware configuration or default settings), or performs some other function such as printing or saving a spreadsheet.

For each spreadsheet program before the court, the number of executable operations is large. Also, the possible methods of presenting available operations to the user is large. One simple approach would be to give each operation a unique name. If this were done, however, the large number of unique names would make it difficult for a user to remember and invoke them expeditiously.

A common way of overcoming this problem is to develop a menu hierarchy. The first level of the hierarchy presents the user with a "menu" consisting of a limited number of "menu commands." Some of these menu commands may be executable operations (*e.g.*, "Quit") causing the program to do something (*i.e*, terminate). When the user invokes an executable command, the program performs the corresponding operation and does not present any further menu options. To proceed farther in the hierarchy, the user must start again and select a different option.

Other menu commands are not in the set of executable operations; instead they are "internal" menu commands, each of which substitutes a new menu in place of the menu in which the internal command appears. Like the menu it replaces, the substituted menu (or "submenu") may consist of a combination of executable commands and internal commands.

The menu commands and menu structure are commonly described as a "tree." The imagery is imperfect, however, unless one thinks of a rather unusual tree that has a leaf or two as well as branches at most junctures. In this imagery, some of the choices at each juncture may be branches and others may be leaves. Each branch or leaf has a name. The "name" of the branch or leaf is a "menu command" within the program's "menu tree". The user starts at the trunk, and by choosing a branch, starts a climb upward. (One may, of course, also envision an upside-down tree, with the user working downward.)

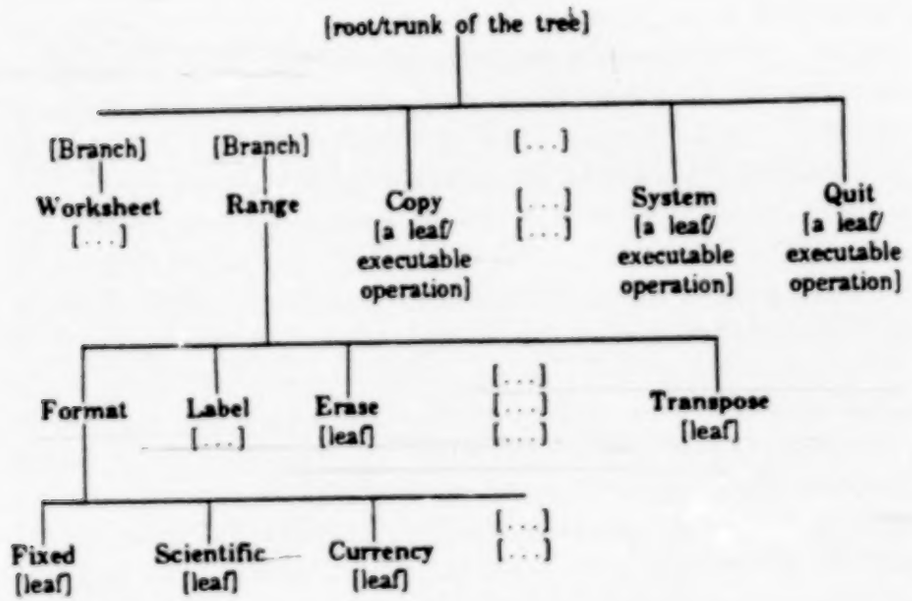
If instead of choosing a branch at the first juncture the user chooses an executable operation, no climb occurs. A name (menu command) corresponding to an executable operation is a leaf of the tree. Having reached that leaf, the user cannot climb farther; no new branches are presented. Instead, the operation is executed. To climb farther (or elsewhere in the tree), the user must go back at least a bit (or all the way to the beginning) and choose branches up to a higher level. As long as the user invokes an internal menu command (a branch) at each juncture, the program presents to the user a new set of

branches and leaves, and by choosing a branch the user climbs higher.

The height of the climb (the number of branches that the user must select before reaching a "leaf" or executable operation) varies depending on which choices the user makes at each juncture. Thus, if "Quit" is selected at the first level, the user makes no climb; the user has selected a leaf, not a branch.

To format a cell as currency data in Lotus 1-2-3, the user climbs three levels up—choosing the range branch, followed by the format branch, followed by the currency leaf.

Switching the imagery to an upside-down tree (a more common convention in computer software discussion), envision a downward progression. For example, the following is a graphic display of the part of the tree a user would climb (or descend) to select the currency operation.



The term "menu tree" refers to the structure of all of the branches and leaves in the menu hierarchy as well as the names of those branches and leaves (*i.e.*, the names of the menu commands).

In the July 31 Memorandum and Order, I concluded that the definition of the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 includes the set of executable operations (leaves) of the tree (although not necessarily the menu command name assigned to each operation). As part of the "idea", the determination of the function of each executable operation is not protected by copyright law.

The menu tree expresses to the program user, in hierarchical fashion, the array of available operations. Even though the executable operations are not copyrightable, the menu tree is copyrightable because the (hierarchical) arrangement of the definition and identification of executable operations contains expression. Accordingly, in resolving the parties' motions for summary judgment, I concluded that many of the details of the menu tree in Lotus 1-2-3 were not a part of the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program, but rather a part of the expression of that idea.

2. *The Lotus and Borland menu tree.*

The original complaint alleges infringement of Lotus's copyright in Lotus 1-2-3 version 2.0 (and earlier versions). In deciding the issues before the court, I have examined both Lotus 1-2-3 version 2.0 and version 2.01. Version 2.01 is a copyrighted derivative work based on version 2.0. Borland points to no differences between these programs that would have a bearing on any issue in this case. A comparison of the menu tree for Lotus version 2.0 (Exhibit 524) and the menu tree accessible through the version 2.01 program also demonstrates no differences material to this litigation. Indeed, the parties earlier stipulated (1) that the menu commands are the same and are displayed in an identical fashion to the user, and (2) that no difference exists between the user interfaces that would have any effect on the issues in this case. *See* O'Connor Dec'n. Docket No. 108, Exh. H. Accordingly, I will

refer to the menu tree generically as the Lotus 1-2-3 menu tree although I have examined both versions 2.0 and 2.01.

The Lotus 1-2-3 menu tree contains approximately 469 menu commands. *See* Kieras Dec'n, V:& at ¶ 66.

> (The record in this case consists of nine volumes of material. Citations to this record are formatted as Volume:Tab; thus "V:8" means Volume V, Tab 8.)

Portions of the menu tree are illustrated at various places in the Exhibits. The entire tree is set out in Exhibit 524, A-C, E (the "Flesher Exhibits" which were originally submitted with the summary judgment materials). As the number of commands suggests, the menu tree is quite large.

The Quattro and Quattro Pro programs have both a native menu tree and a "1-2-3 emulation" menu tree. I do not understand Lotus to contend that the native menu tree infringes any Lotus copyright. Accordingly, when I refer to the Quattro or Quattro Pro menu trees, I refer only to the menu tree used in the program's 1-2-3 emulation interface.

The Quattro menu tree contains approximately 3370 menu commands. (Kieras Dec'n, V:8 at ¶ 77.) This menu tree is illustrated in Flesher Exhibit B. The Quattro Pro version 1.0 menu tree contains approximately 5216 commands (*Id.* at ¶ 66.) This menu tree is illustrated in Flesher Exhibit A.

The Quattro Pro program has been released in several versions, specifically versions 1.0, 2.0, 3.0, 4.0, 4.01, SE, and Quattro Pro for Windows. Only versions 1.0, 2.0, 3.0, and 4.0 contain the emulation interface. In deciding issues for Phase I of the trial, therefore, when I refer, without further specification, to the Quattro Pro menu tree, I refer only to the emulation menu tree in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0.

The entire Lotus menu tree is contained within both the Quattro and Quattro Pro version 1.0 menu trees. For example, the first menu in Quattro Pro version 1.0 contains an identical copy of the 1-2-3 menu commands, in the same order, but with one new command inserted ("View"). Invoking a Borland command that is identical to a Lotus command produces a menu that is an identical copy of the Lotus submenu, but (in

some cases) with one or two new commands inserted. Thus, each menu or submenu in Lotus 1-2-3 is reproduced identically, but with the insertion in the Quattro and Quattro Pro menu trees of some new menu commands and any submenus associated with the new menu commands.

Put another way, both the Quattro and Quattro Pro version 1.0 menu trees consist of a *virtually identical* copy of the entire 1-2-3 menu tree, with new branches or leaves inserted at various places.

Although the above comparison focuses only on version 1.0 of the Quattro Pro menu tree. the scope of copying remains the same for versions 2.0, 3.0, and 4.0. The only difference among the emulation menu trees in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0, is that each successive version added new commands. *See* Warfield Dep'n, III:40 at 72-73. Thus, later versions also contain a virtually identical copy of the Lotus menu tree, with still more new branches or leaves added.

The presentation of the commands in the Quattro and Quattro Pro programs indicates which commands are found in the 1-2-3 menu tree and which are Borland's insertions. In Quattro, the new (and only the new) menu commands are followed by a dot at the place where the new command is inserted into the tree. Thus, in Quattro, the "Install" command was inserted into the first level menu followed by a dot. The submenu of "Install" does not have these same dot designations. Such a designation is unnecessary for this submenu, however, because everything within the newly inserted "Install" branch of the tree is new.

Quattro Pro uses the same approach for differentiating 1-2-3 menu commands and added menu commands, with the exception of "View" in Quattro Pro's first menu, which has no designation as an inserted command but whose submenu consists of commands each followed by a dot. In addition, in the Quattro Pro menus, a line generally appears separating added menu commands from original 1-2-3 menu commands.

The disparity in total number of menu commands between the 1-2-3 menu tree and the Quattro and Quattro Pro menu trees is large, but does not alter the scope of copying. Most of

the new commands are clustered together far up in the tree (or down in the upside down tree). For those menus that exist in the Lotus 1-2-3 menu tree, the number of added menu commands is rarely more than one or two. The effect is similar to an identical copy of a book with some paragraphs and lengthy footnotes inserted, and some voluminous appendices attached at the end.

In short, I adhere to my earlier conclusion that no reasonable factfinder could find that Borland did not take from Lotus 1-2-3 the menu commands and menu command structure substantially as they were. Further, I find that Borland produced a virtually identical copy of 1-2-3 menu tree, albeit with additions, in its Quattro and Quattro Pro emulation interfaces.

B. Nature of the Work—Scope of Illicit Copying.

1. Scope of limitations imposed by functional considerations.

a. The functional considerations.

Borland advanced the following "constraints" on design of a menu tree.

(1) Each menu command was chosen to tell the user the purpose of the menu command and its function.

(2) Each menu command was selected so that it had a different first letter from the other menu commands within the same menu.

(3) Each menu was set up to have only about seven choices (witnesses referred to this as the "seven plus or minus two rule," *i.e.*, no menu should have fewer than five or more than nine commands).

(4) The menus were structured so that words within the menu that dealt with similar functions were grouped together.

(5) Executable operations that were likely to be frequently used were located near the top of the (upside down) tree.

(6) Menu commands within a menu were arranged from left to right in order of decreasing frequency of use.

(7) Commands in submenus were grouped under the menu command to which they relate.

See VII:A:6. Borland also proffered an eighth functional "constraint" of having each menu fit on one line of a computer screen. If this constraint were followed, each menu could have no more than 80 characters (the number of characters across a standard computer terminal screen).

Borland refers to the above items as "constraints" on design of a menu tree. Lotus contends that they are "guidelines" or "rules of thumb." The difference in terminology is largely if not entirely semantic. Nevertheless, the different terms raise different images. "Constraint" implies a rule that must not be violated. If "constraint" is defined this way, the eight listed items are not "constraints." Each of them is violated somewhere in the Lotus 1-2-3 menu tree (except possibly the second, which has been violated in other menu trees, *e.g.*, Excel 3.0 with "*S*ave" and "Save *A*s" in the same menu, and the last, which was violated in SuperCalc). Moreover, the alleged "constraints" are full of terms that offer little guidance on how to conform. For example, when is one menu command "similar" to another? The answer may be easy in some cases and entirely within the discretion of the programmer in others.

"Guideline," or "rule of thumb," is a more accurate description. Each of these terms implies a directive that is not always to be followed. Even these terms, however, ordinarily imply a precise rule. The first item listed above is hardly precise, even though it does limit the possible forms of a menu tree. Accordingly, I will refer to these as functional "considerations."

b. Impact of functional considerations on form of the menu tree.

The parties dispute the impact of the eight functional considerations (listed above) on the freedom of expression in forming a menu tree. For the following reasons I conclude that the Lotus 1-2-3 menu tree is just one of a great variety of possible expressions that are consistent with the functional considerations listed above and the specific set of executable operations used in Lotus 1-2-3.

First, none of the proffered functional considerations is overriding or dictates any specific result. Each may be violated. Most are violated at one or more points within the Lotus 1-2-3 menu tree. Indeed, the considerations are often competing and must be traded off against each other. *See, e.g.*, Gottheil Dep'n, IX:11 at 143-64; Olson Dep'n, III:32 at 117-18. No functional considerations are offered for guiding the determination of how to trade off competing concerns. Borland provided no credible evidence explaining how functional considerations could completely control formation of the menu tree.

Second, the "quality" of the menu tree depends on the peculiarities (*i.e.*, the particular tastes) of the individual using the program. *See, e.g.*, Bosworth Dep'n, I:1 at 389-90. For this to be true, there must be a variety of possible menu trees for the user to choose from. In this regard, Borland programs offer a "menu builder" that enables users to alter, customize, and create menu trees. If functional considerations restricted the possible expression of the menu tree to a limited number of possibilities, there would be little or no need for a user to modify it.

Third, even Borland's experts acknowledge that, given all of the various functional considerations, at least a limited range of choices remains for individual menu commands. *See* Liddle Dep'n, II:27 at 121; Olson Dec'n, V:13 at ¶ 35. Thus, "Copy" could be called "Replicate," "Duplicate," "Reproduce," "Repeat," "Ditto," etc. "Range" could be "Block," "Scope," "Extent," "Cells," etc. Although some experts contend that certain words and menu structure are preferable to

others, these same experts contend that the words Lotus selected did not matter for 1-2-3's success. *See* Liddle Dec'n, V:11, ¶ 55; Olson Dec'n, V:13, ¶¶ 36, 38. In any event, even if there were as few as two acceptable words for each menu command (given the other functional considerations), there would be 2 raised to the 469th power possible menu trees (an astronomical number) having precisely the same menu structure as the 1-2-3 menu tree, but with variations in menu command names. *See* Emery Dec'n, V:3, ¶ 85. Even if only half of the menu commands had more than one possible name (given the other functional considerations), there are over 2^{234} possible menu trees having the same menu structure.

In listing these examples, I do not mean to suggest that the alternatives for menu command names are so few, nor to imply that every synonym for a command word is suitable in view of the other functional considerations. Rather, I merely note the breadth of possible menu trees that may be achieved in this manner as a factor bearing upon whether implementation of the idea, system, process, procedure, or method underlying the Lotus 1-2-3 menu tree is capable of a wide variety of expression.

Also, in presenting this analysis, I do not imply that counting possible variations in individual words is decisive of the breadth of possible expressions for literary works in general. It would not necessarily be significant, for example, to determine how many ways instructions for a simple game can be expressed solely by examining possible alternatives for individual words. Nevertheless, variations of the words in the menu tree represent a material consideration in the context of this case because of (1) the nature of the menu tree, which unlike an English sentence, permits substitution of individual words without changing the meaning of the menu tree, (2) the size of the menu tree, and (3) the structure of the menu tree.

If the designer of a menu tree chooses not to copy the structure of Lotus 1-2-3, the designer's freedom of expression and range of possible expressions for the menu tree expand dramatically. *See* Emery Dec'n, V:3, ¶ 84. Nothing in the materials before me supports an inference that functional considerations alone control the structure of the menu tree.

Indeed, Borland's experts implicitly acknowledge alternatives for the structure of the 1-2-3 menu tree. *See* Liddle Dec'n, V:11, ¶ 56; Olson Dec'n, V:13, ¶ 38.

Finally, a number of other spreadsheet programs use vastly different menu trees. The existence of vastly different menu trees in other commercial programs supports the conclusion that the Lotus 1-2-3 menu tree is but one of many possible forms for a menu tree.

Borland contends, however, that the differences in menu trees between programs such as Lotus 1-2-3 and Excel are due to differences in the programs' functionality. That is, Borland argues that because the executable operations (leaves of the tree) are different and the visual displays are different (*e.g.*, pop-up menus vs. text on a single line), the menu trees are different. Borland contends that, as a result, the existence of other commercial programs with menu trees vastly different from the 1-2-3 menu tree (and each other) is not probative of the degree of freedom of expression that exists in formulating the 1-2-3 menu tree.

I reject Borland's contentions for the following reasons. Although differences in program functionality may explain some differences between menu trees, the differences in functionality cannot explain the breadth of differences among menu trees used in the various programs. For example, functional considerations do not explain why the "File" menu command is left-most in the Excel menu tree and fifth from the left in the 1-2-3 menu tree. Borland does not suggest or offer any evidence explaining how the (unspecified) differences in functionality among 1-2-3, Excel, and Quattro's native mode affect placement of the "File" command within the first menu. Similarly, 1-2-3 places the copy and move command in the first level of the menu tree. In Excel they are in the second level.

The differences among menu trees in the various programs submitted are so large that they are, in a practical sense if not literally, incapable of enumeration. The broad scope of these differences cannot be explained in terms of differences in functionality. Indeed, Borland offers no evidence or argument providing a reasoned explanation (as opposed to an unsupported assertion) of how the magnitude of differences could be explained by any differences in functionality. I conclude that many of the differences are due to different choices about how to express to the user the available user choices about all the particular operations that the program can perform.

In argument, Borland placed primary emphasis on an expert's assertion that the set of possible menu command names is "not large." *See* Olson Dep'n, III:32 at 54-59. As a preliminary matter, I find that this testimony is entitled to little, if any, weight. Nothing purports to explain how the expert reached this conclusion. In fact, this expert admits that she has never attempted to determine how large the set is. An expert's unexplained and unsupported assertion, even if received into evidence, is scant basis for reaching a reasoned finding.

In addition, the testimony is unclear whether the expert meant that there was a small set of choices for individual command names, or the set of all menu command names. In an earlier declaration this expert conceded that there is a "very narrow range" of suitable words for individual commands. *See* Olson Dec'n, V:13 at ¶¶ 35-36. As observed above, even if (given all the functional considerations) only half the commands have only one alternative label, the possible number of menu trees that differ in detail remains very high. Consequently, what this expert meant by "not a large set" may be entirely consistent with a determination that the number of possible ways of expressing the menu tree is without limitations that are material to the separation between idea or function and expression.

Finally, even assuming that the alternatives for menu command names were few, when I take into account as well the options available for structure, I find that the number of possible choices again expands dramatically. Thus, what Borland copied from 1-2-3 was not limited to aspects dictated by functional considerations. Rather, Borland copied the entire menu tree, much of which was the free expression of the creators of Lotus 1-2-3.

c. Conclusion

As Borland practically conceded in closing argument, application of functional considerations does not restrict the expression of the menu tree to essentially only one form. (*See* Docket 333 at 3-52.) Further, I find that although functional considerations may have some effect on the design of a menu tree, they do not impose any practical limitation on the possible forms of expression to a number far enough short of infinity that any way of expressing the number in English words has come into common usage. The set of executable operations in Lotus 1-2-3 is large and the possible structural variations are enormous. The menu tree is capable of a very wide variety of expressions.

Borland has not argued or provided any evidence that any specific aspect of menu structure or command names, short of the entire menu tree, is dictated solely or influenced mainly by functional considerations. Moreover, Borland has used a virtually identical copy of the 1-2-3 menu tree. Accordingly, this case on its facts does not present any issue that might arise on a finding of copying of something short of virtually the entire menu tree.

2. Originality.

In its closing argument, Borland asserted that, because of functional considerations, creation of the menu tree did not require sufficient originality to justify protection under copyright law.

a. Originality doctrine.

The Supreme Court recently addressed the requirement of originality in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The Opinion for the Court in *Feist* observed that the requirement that a work be "original" derives from the Constitution. The Court described the originality requirement as follows.

> The *sine qua non* of copyright is originality. To qualify for copyright protection, a work must be original to the author. Original, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity. To be sure, the requisite level of creativity is extremely low; even a slight amount will suffice. The vast majority of works make the grade quite easily, as they possess some creative spark, no matter how crude, humble or obvious it might be.

Id., 499 U.S. at ___, 111 S.Ct. at 1287 (internal quotation marks and citations omitted).

The Court held that copying from the white pages of a telephone book was permissible. To reach this conclusion, the Court first confirmed the firmly established principle that facts (*e.g.*, the names, towns and telephone numbers in the white pages) are not copyrightable. The Court also confirmed that "even a directory that contains absolutely no protectable written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection or arrangement." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1289. In this circumstance, "the copyright is limited to the particular selection or arrangement." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1290.

Applying these two basic principles, the Court determined that the selection and arrangement of facts in the plaintiff's telephone book, *i.e.*, selection of all people who applied for phone service and arrangement in alphabetical order, was "devoid of even the slightest trace of creativity." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1296.

> [O]riginality is not a stringent standard; it does not require that facts be presented in an innovative and surprising way. It is equally true, however, that the selection or arrangement of facts cannot be so mechanical or routine as to require no creativity whatsoever.

Id. Indeed, the selection of all people and placement in alphabetical order was not at all original to the plaintiff—phone books had been arranged in that fashion for years. Plaintiffs

white pages were nothing but a "garden-variety" phone book. *Id.*

b. Originality in this case.

Relying on *Feist*, Borland contends that the eight "rules" regarding functionality listed above demonstrate that the form of the menu tree (including menu command names and structure) is not original. For the following reason, I find that the menu tree is "original."

As explained above, the menu tree of Lotus 1-2-3 may be viewed as an arrangement of the executable operations as "leaves" of the "menu tree," with internal commands ("branches") leading through the structure to the "leaves." Altogether, the tree contains 469 labels to explain the arrangement. This text and arrangement presents the set of executable operations to the program user. Thus, the specific operations could be viewed as (uncopyrightable) facts and the menu tree as an arrangement (with textual labels) of the facts. Even viewed in this light, most favorably to Borland, the menu tree easily satisfies the originality requirement discussed in *Feist*.

The functional considerations Borland propounds are not at all comparable to application of a rule of alphabetical ordering of executable operations.

Borland's "rules" are violated with regularity in the 1-2-3 menu tree. In contrast, the rule of alphabetical ordering was not violated in the *Feist* white pages.

Also, the functional considerations listed above were not "an age old practice, firmly rooted in tradition and common place." *Feist*, 499 U.S. at ___, 111 S.Ct. at 1297.

Moreover (except in a few places where rules of alphabetical ordering differ) one who is given the names, addresses and numbers in a phone book, can generate the alphabetical ordering by routine and mechanical application of rules, with only one possible outcome. In contrast, the text for the menu commands and the menu structure itself are not dictated by mechanical application of the functional considerations. As noted above, a wide variety of menu trees is possible.

In sum, the 1-2-3 menu tree "make[s] the grade quite easily, as it possess[es] some creative spark. . . ." *Feist*, 499 U.S. ___, at 111 S.Ct. at 1287.

Borland nevertheless contends that, because functional considerations played a role in formulating the 1-2-3 menu tree, the menu tree is not copyrightable. Borland's argument is susceptible of two interpretations, both lacking merit.

First, one may interpret this argument as a contention that the functional considerations so permeate formulation of the menu tree that the menu tree is not separable from the "idea" of the program. This form of argument has nothing to do with the amount of creativity or originality involved. Derivation of a scientific formula may require a great deal of creativity and produce an original result. If the formula fails the copyrightability test, it is because the formula is not expressive—independently of creativity or originality. Casting the argument in terms of originality doctrine rather than separability does nothing to assist resolution of the issues in this case, and may lead to confusion. In any event, to the extent Borland raises the same separability argument that it has raised before, but dressed now in terms of originality, that argument was previously rejected by the court for good reason, and is now rejected again.

Alternatively, Borland may be understood as contending that any work whose form is restricted to any material extent by functional considerations is not original. Without more, this contention is invalid on its face. Any original literary work is formulated according to functional considerations imposed by language, a desire for clarity, and a desire to express the ideas conveyed. As one of Borland's experts concedes, the first functional consideration (conveying the nature of executable operations) is not materially different from a functional consideration for selection of words in any English writing. *See* Olson Dep'n, III:32 at 54. Accordingly, this type of "functional consideration" can remove a writing from copyright protection only if it restricts the forms of expression (that are separable from the idea or function of the work) to a limited number. Thus, cases referring to functional con-

siderations and decided under originality doctrine, including those cited by Borland, uniformly refer to limits on the number of forms of expression given functional considerations.

> *See Victor Lalli Enter., Inc. v. Big Red Apple, Inc.*, 936 F.2d 671, 673 (2d Cir.1991) ("purely functional grids that offer no opportunity for variation"); *Sinai v. California Bureau of Automotive Repair*, 25 U.S.P.Q.2d 1809 [1992 WL 470699] (N.D.Cal.1992) ("limited number of ways" to arrange information in chart); *New Haven Copper Co. v. Eveready Mach. Co.*, 229 U.S.P.Q. 838 [1986 WL 269] (D.Conn.1986) (column headings "dictated by functional considerations"); *Merritt Forbes & Co. v. Newman Inv. Sec., Inc.*, 604 F.Supp. 943, 951-52 (S.D.N.Y.1985) (no originality if "form of expression is dictated solely by functional considerations"; determining that fact issue exists over whether concept is "capable of varied expressions"); *Decorative Aides Corp. v. Staple Sewing Aides Corp.*, 497 F.Supp. 154, 157 (S.D.N.Y.1980) (what was similar was "dictated by functional considerations"), *aff'd without op.*, 657 F.2d 262 (2d Cir.1981). *Cf. Feist*, [499 U.S. at ___], 111 S.Ct. at 1296 (routine and mechanical application of single rule).

Thus, Borland's argument fails because, as above, a wide variety of expression is possible for the Lotus 1-2-3 menu tree.

C. Value of Expert Testimony.

In reaching the conclusions above, I have read and weighed the expert testimony offered by the parties. For the following reasons, however, much of this testimony has little weight in relation to the issues of law and fact that are decisive of the outcome in this case.

First, most of the testimony expresses conclusions, without any reasoned explanation of the basis for the conclusions. Second, much of the testimony uses terms such as "system" or

"a large number" that the experts do not define. As factfinder, I choose not to abandon my factfinding responsibility by accepting an expert's proposed unexplained choice. This is my usual practice, and for even stronger reasons I proceed in this way when the expert's testimony uses terms specially defined by the expert to have meanings that incorporate into the terminology substantive choices that have important policy implications, without examining and explaining reasons for the policy choices and whether they are consistent with applicable statutes and precedents.

Finally, much of the testimony is based on explicit or implicit assumptions about copyright law that are incorrect. For example, one expert testified that she was unsure of whether design of a menu tree is creative. In defining creativity, however, the expert explained that creativity required some "inventive leap." Olson Dec'n, III:32 at 124-35. Of course, this type of creativity—the inventive leap or new idea—is not required for copyrightable expression. *See Feist*, 499 U.S. at ___, 111 S.Ct. at 1296 (novelty or innovation not required). Expression is copyrightable, even when it is a new expression (*e.g.*, a new novel about young love) of an old idea (*i.e.*, boy meets girl).

D. Conclusion.

In sum, I conclude that each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression.

III. Affirmative Defenses for Phase I.

A. Laches.

Borland has the burden of proving laches. *Costello v. United States*, 365 U.S. 265, 282, 81 S.Ct. 534, 543, 5 L.Ed.2d 551 (1961). A defense of laches has been established if the defendant proves that (1) the plaintiff inexcusably or

unreasonably delayed in bringing an infringement action, and (2) defendant was prejudiced by this delay. *See id. See also* 3 Nimmer on Copyright § 12.06.

The chronology of the principal events bearing upon Borland's defense of laches commences in January 1987, when Lotus filed suit in the *Paperback* case. Borland was developing Quattro at the time. Before including the 1-2-3 emulation interface in Quattro, Borland secured a legal opinion that the program did not infringe Lotus's copyright. In late September 1987, Borland announced Quattro, Borland's first spreadsheet product. In November, Borland shipped the Quattro product. Quattro was advertised widely, one advertisement appearing in a November 1987 issue of "Lotus Magazine."

In early 1988, the Quattro product was discussed at a high-level Lotus meeting. The evidence concerning this meeting supports an inference that certain Lotus executives discussed the possibility that Borland would change Quattro if Lotus threatened suit. The evidence does not prove, however, that Lotus delayed filing suit or kept silent on the issue of infringement *for this reason*. I find that Lotus may have discussed the possibility that Borland would change its product, but that this was not a factor in Lotus's decision not to file suit in 1988.

Some time after the meeting, Lotus decided not to file suit against Borland then. In addition, Lotus adopted a policy of not commenting on possible infringement by, or legal action against, Borland.

In September 1988, less than one year after Borland shipped Quattro, Borland acquired the "Surpass" spreadsheet technology (from "Surpass Software Systems") for approximately $2.4 million. This included a software program that had been called "Surpass." The Surpass program's only menu tree was a copy of the 1-2-3 menu tree. After acquiring Surpass, Borland removed the Surpass product from the market. Borland then re-engineered Surpass to become the Quattro Pro product, having a 1-2-3 compatible menu tree as just one of the possible menu trees available. Quattro Pro version 1.0 was released in November 1989.

Near the end of June 1990, this court handed down its decision in the *Paperback* case. The day after the decision was released, Borland filed an action in California seeking a declaration of noninfringement. On July 2, 1990, Lotus filed the original complaint in the present action.

1. Delay.

The period between Borland's release of Quattro and Lotus's filing of the instant action was approximately $2^1/_2$ years. This amount of time, standing alone, does not dictate a finding that the delay was unreasonable.

> *Cf. Roulo v. Russ Berrie & Co.*, 886 F.2d 931, 942 (7th Cir.1989) ("A two year delay in filing an action following knowledge of the infringement has rarely been held sufficient to constitute laches."), *cert. denied*, 493 U.S. 1075 [110 S.Ct. 1124, 107 L.Ed.2d 1030] (1990); *Hoste v. Radio Corp. of America*, 654 F2d 11 (6th Cir.1981) (per curiam) (error to grant summary judgment based on laches for suit filed after 13 year delay); *Boothroyd Dewhurst, Inc. v. Poli*, 783 F.Supp. 670, 680 (D.Mass 1991) (four year delay may not constitute laches).

Whether this amount of delay was unreasonable or inexcusable depends on the motives for delay. Thus, Judge Learned Hand observed that

> it is inequitable for the owner of a copyright, with full notice of an intended infringement, to stand inactive while the proposed infringer spends large sums of money in its exploitation, and to intervene only when his speculation has proved a success. Delay under such circumstances allows the owner to speculate without risk with the other's money; he cannot possibly lose, and he may win.

Haas v. Feist, 234 F. 105, 108 (S.D.N.Y. 1916). On the other hand, delay used to evaluate and prepare a complicated claim may be reasonable. *See, e.g., Paperback*, 740 F.Supp. at 82.

Similarly, delay to determine whether the scope of proposed infringement will justify the cost of litigation may be permissible. *See, e.g., Boothroyd Dewhurst*, 783 F.Supp. at 680-81. In the circumstances of the present case, I find that Lotus's delay in filing suit was reasonable and excusable.

First, I reject Borland's contention that Lotus's delay in bringing suit is of the kind addressed in *Haas, i.e.*, in order to speculate with Borland's money. Borland relies primarily on the failure of Lotus to file an action in the early part of 1988. Borland contends that Lotus then knew enough about Borland's plans that Lotus should have notified Borland immediately if Lotus intended to claim infringement and that Lotus did not do so because Lotus was afraid that Borland would change Borland's product. Borland contends, in other words, that Lotus waited to file suit in an effort to trap Borland into expending large sums of money on Borland's spreadsheet products and, later, to reap the benefits of Borland's investment.

I find, however, that the evidence Borland presented does not support Borland's contentions. At the time of the 1988 meeting, Borland had already developed, advertised, and shipped Quattro. The possibility that Borland would adjust Quattro to remove the copy of the Lotus 1-2-3 menu tree was marginal in significance at the time of this meeting. For this reason, reinforced by the findings stated above, I do not find the evidence sufficient to persuade me, as factfinder, to draw the inference that Lotus waited to sue hoping that its *competitor's* product would succeed and enable Lotus to reap enhanced damages.

> *See Russell v. Price*, 612 F.2d 1123, 1126 (9th Cir.1979) (conclusion that plaintiff was not speculating on defendant's success supported by fact that plaintiff was also seeking to exploit the copyrighted material), *cert. denied sub nom., Drebin v. Russell*, 446 U.S. 952 [100 S.Ct. 2919, 64 L.Ed.2d 809] (1980).

I find that Lotus delayed bringing suit for the purpose of awaiting resolution of the *Paperback* and *Mosaic* cases.

Throughout 1988, Lotus was involved in extensive litigation in *Paperback* and *Mosaic*. This litigation was hotly contested. If Lotus had lost the *Paperback* case, Lotus's claims against Borland would have been difficult to maintain. Thus, by waiting to file suit, Lotus avoided expensive duplicative litigation all of which might have been unsuccessful if Lotus lost in *Paperback*. This avoided a risk of needlessly wasting court and party resources. Once the *Paperback* case was resolved by the trial court, Lotus filed suit against Borland almost immediately. I cannot say that this was unreasonable.

Borland contends, nevertheless, that Lotus's delay was unreasonable in part because Lotus kept silent about its belief that Borland's product infringed. Lotus responds that it was satisfied that Borland knew of the infringement claims because of (1) Phillipe Kahn's (Borland's chairman's) strong public reaction condemning Lotus for filing the *Paperback* case, and (2) a public statement Lotus issued before Quattro was shipped that appeared in the May 26, 1987 issue of PC Magazine (Exh. 28 at 162-63). In the public statement, Lotus represented that it did not claim copyright over the ideas of a two-line moving cursor interface or context sensitive help. Lotus then characterized its suit against Paperback and Mosaic as protecting against companies that copied "all of the ways in which 1-2-3 communicates to the user, including its menu structure and sequence, word selection, and macro language design. . . ."

I find that Lotus reasonably believed that Borland was aware of Lotus's potential claims against Quattro and Quattro Pro. Borland "was far too sophisticated to need being led by the hand." *Famous Music Corp. v. Bay State Harness Horse Racing & Breeding Assoc., Inc.*, 554 F.2d 1213, 1215 (1st Cir.1977) (rejecting estoppel defense). In light of this, I cannot say that Lotus's decision not to comment on the infringing nature of Quattro or Quattro Pro makes Lotus's delay in this case unreasonable or inexcusable.

In weighing equities, some courts decline to permit a laches defense by a "deliberate pirate," *Haas*, 234 F. at 108, or less harshly stated, a "deliberate infringer."

> *See* 3 Nimmer on Copyrights, § 12.06 at 12-107 (delay "may not be a bar against one who knew of plaintiffs asserted rights, or as against a deliberate infringer") (footnotes omitted).

I find that Borland (1) knew of Lotus's copyright, (2) knew that Lotus was acting to protect that copyright, and (3) deliberately introduced a virtually identical copy of the Lotus 1-2-3 menu tree into its 1-2-3 emulation interfaces.

The fact that Borland did secure a noninfringement opinion from a prestigious law firm is relevant, of course. Nevertheless, Lotus is not in any way accountable for that opinion. Indeed, Borland offers no evidence that Borland gave any notice to Lotus of Borland's seeking legal advice or of the substance of that advice. It is a rather curious twist of argument to suggest that a copyright owner has a duty of notice to the infringer when the infringer has enough concern about its own actions that it seeks a legal opinion privately but refrains from giving any kind of notice to the copyright owner.

Borland contends that the *Paperback* suit could not constitute notice that Lotus was enforcing its copyright because the issues in *Paperback* and this case are now identical. The fact that the issues are not identical, however, is far from decisive. More to the point is the fact that Borland was aware of Lotus's assertion of copyright protection for the 1-2-3 program and in particular for the form of its menu tree.

Borland next argues that Lotus had not sufficiently formulated its precise contentions of what aspects of Borland's products infringe Lotus's copyright until May of 1992—almost two years into this case. Again, this has little to do with the fact that Borland was aware of Lotus's assertion of copyright protection for Lotus 1-2-3 and in particular for the form of its menu tree.

2. Prejudice.

The defense of laches requires that defendant suffer prejudice. Typical forms of prejudice include: death or unavailability of an important witness, dulling of memories, loss of relevant records, and continuing investments and outlays by

the alleged infringer in connection with the operation of its business. *See Eisenman Chemical Co. v. NL Indus., Inc.*, 595 F.Supp. 141, 147 (D.Nev.1984). Borland argues only the last form of prejudice. To demonstrate prejudice, Borland points to marketing costs and acquisition of the Surpass technology.

To constitute prejudice, however, these expenses must have been incurred *as a result* of Lotus's delay in bringing suit. Where an infringer was aware of a plaintiff's copyright, as Borland indisputably was, courts have phrased this point as a requirement that the actions be taken in reliance on the plaintiff's delay in bringing suit.

> *See Russell*, 612 F.2d at 1126; *In Design v. Lauren Knitwear Corp.*, 782 F.Supp. 824, 831 (S.D.N.Y.1991); *Steinberg v. Columbia Pictures Indus., Inc.*, 663 F.Supp. 706, 716 (S.D.N.Y.1987).

For the following reasons, I conclude that Borland has not proved that it took any action in reliance on Lotus's delay in bringing suit.

First, I find that Borland would have invested in and entered the spreadsheet market whether or not Lotus delayed in bringing suit. Phillipe Kahn testified that the decision to include the 1-2-3 emulation interface in Quattro was made not long before the product shipped. (I:14 at 362). Of course, this initial decision to include the 1-2-3 emulation interface was made before Lotus could have objected.

The Surpass technology, acquired in 1988, was used to form the basis for Quattro Pro. This was less than one year after Borland shipped Quattro. Borland presented no evidence that it would not have acquired Surpass had Lotus taken action earlier. Moreover, Quattro already had a 1-2-3 emulation interface. It would not be reasonable to infer that other aspects of Surpass than its copy of the Lotus 1-2-3 menu tree had no influence on Borland's decision. I find that Borland has not proved that it acquired the Surpass technology in reliance on Lotus's delay in bringing suit.

I observe also that Borland filed a declaratory judgment action immediately after the decision in the *Paperback* case.

Borland seeks to explain the timing of the filing of the declaratory judgment action by presenting evidence that Borland was responding to recent conflicting reports that Lotus would file suit against Borland. Even if I credited this self-serving assertion, unsupported by any objective evidence, however, I would not infer that Borland had ever relied on Lotus's failure to assert infringement to conclude that Lotus would never file suit. Moreover, a senior Borland official averred that the Borland Board of Directors had been informed before the decision in the *Paperback* case that Lotus intended to sue Borland if Lotus prevailed in *Paperback*. (Leyton Dec'n, VI:10, ¶ 5.) Nevertheless Borland, like Lotus, awaited the outcome of the *Paperback* case. Significantly, when Borland filed, it sought a declaration of noninfringement. but omitted any specific request for declaration of unenforceability due to laches or estoppel.

Based on the evidence before me. I find that Borland developed and marketed Quattro and Quattro Pro in reliance on (1) the noninfringement opinion, and (2) the hope that Lotus would lose the *Paperback* litigation. Further, I find that Borland has not proved that it took any actions in reliance on Lotus's delay in bringing suit Finally, I find that reliance, even if it had occurred, would have been unreasonable in the circumstances shown by the evidence in this case.

B. Estoppel.

The parties apparently agree that, to establish a defense of estoppel, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. *See Precious Metals Associates, Inc. v. Commodity Futures Trading Comm'n*, 620 F.2d 900, 908 (1st Cir.1980). In view of this agreement, I need not and do not consider whether this interpretation of *Precious Metals* and other precedents is more generous to Borland's estoppel claim than a more precise understanding of the precedents would be.

To establish estoppel, Borland relies primarily on Lotus's delay in bringing suit and Lotus's failure to inform Borland of Lotus's belief that the Borland products infringe. For the reasons explained in Section A above, I find that (1) Lotus did not intend that Borland infer from Lotus's silence that copying of the menu tree was permissible, (2) Borland was aware of Lotus's copyright, (3) Borland did not rely on the silence of Lotus to Borland's detriment, and (4) any reliance by Borland in the circumstances of this case would have been unreasonable and unjustifiable.

Borland also offers a collection of actions or statements by Lotus that Borland contends entitled Borland to infer that Lotus would not sue Borland for copyright infringement. Borland cites a January 1987 InfoWorld article, released before Lotus had any reason to know of Borland's Quattro product. The article purports to quote a Lotus spokesman, Greg Jarboe:

> "Some folks have misinterpreted that what we have done is to copyright the spreadsheet or the two-line [command] interface, neither of which we are trying to do," Jarboe said. The suit was targeted at two vendors who had "copied 99 percent" of a Lotus product, he said, adding that the company is considering issuing a position statement to clarify the limits of its copyright.

InfoWorld, January 26, 1987. Even assuming this is an accurate quote, something Mr. Jarboe denies, I find that it would not be reasonable to launch a spreadsheet product in reliance on this statement. First, as Borland's general counsel concedes, II:20 at 272, the statement does not purport to disavow filing suit against future products that infringe Lotus's copyright in a manner different from the *Paperback* suit. A characterization of the *Paperback* case as involving "clones" is not reasonably interpreted as expressing an intention not to sue other infringers as well as makers of "clones." Second, the article refers to an upcoming position statement (the PC Magazine statement quoted in Section III.A, above). Although Borland contends that none of Borland's responsible representatives saw this position statement (a contention that I find

questionable in light of their other activities), their failure to see and take account of that position statement, along with acting on the assumption that no such statement had been made, was unreasonable. Thus, to the extent Borland claims that it relied on the InfoWorld statement, it was unreasonable for Borland to rely on that statement while failing to take further steps to determine what Lotus's position statement contained.

Borland next cites the fact that Lotus Magazine published advertisements for Borland's products. I find that it would not be reasonable for Borland to assume that the acceptance of the advertisement in Lotus Magazine showed that Lotus had determined that it would make no infringement claim against Borland. Following Borland's line of reasoning, one would conclude that by allowing competitors to run advertisements in Lotus Magazine. Lotus Development Corporation agreed that the competitor's advertising claims were correct (*e.g.*, that Borland's products are superior, as the advertisements implied) and that Lotus wanted individuals to purchase from Lotus's competitors. Merely stating this chain of inference is enough to expose its unreasonableness. I find, instead, that the fact that Lotus Magazine accepted the Borland advertisement indicated a policy of accepting advertisements—even from competitors. Moreover, on the evidence before me I cannot find, as factfinder, that the plaintiff in this case, Lotus Development Corporation, controlled the decision-making process about what appears in Lotus Magazine. Accordingly, it was not reasonable for Borland to infer that Lotus Development Corporation was making any statement concerning possible copyright infringement when Lotus Magazine accepted the advertisement.

Borland next cites examples of occasions where Lotus personnel interacted with Borland personnel at trade shows and with respect to new products and services, without mentioning any belief that the Borland products infringe. Nothing in these materials suggests that the Lotus personnel involved had the actual or apparent authority, or even sufficient knowledge, to comment on whether or not Lotus believed that Borland

infringed a copyright or whether Lotus intended to file an action in the future. Moreover, nothing in the record shows that the individuals involved actually said anything that would support an inference by Borland that copying of the menu tree was permissible. Accordingly, I find that Borland could not reasonably interpret these communications to mean that Lotus would not file suit.

In sum, I find that both separately and taken as a whole, the events, conduct, and communications that Borland points to in an effort to bolster claims for estoppel fall far short of supporting estoppel.

C. Conclusion.

For the reasons explained above, I find that Lotus's claims in Phase I of the trial are not barred by laches or estoppel. In presenting its case on the affirmative defenses, Borland offers a number of proposed inferences that this court might draw, based loosely on evidence before the court. Although I have not explicitly referred to each of Borland's individual contentions in this Opinion, I have examined all of the contentions and have found them without merit.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

July 31, 1992

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, Mass., Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, Mass., Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, Cal., David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, Cal., Peter Erich Gelhaar, Donnelly, Conroy & Gelhaar, Boston, Mass., Gary L. Reback, Wilson Sonsini, Goodrich & Rosati, Palo Alto, Cal., *for defendant.*

MEMORANDUM AND ORDER

KEETON, *District Judge.*

By Memorandum and Order of March 20, 1992, the court dismissed the parties' motions for summary judgment in this copyright infringement action and invited new motions compatible with rulings therein announced. Each party has renewed its motion for summary judgment and filed further

submissions (Docket Nos. 168-190).* A hearing on these motions was held on May 19, 1992, and additional submissions were filed after that hearing.

The reader may find background information in two earlier documents issued by this court: the first, the Opinion in a related case involving the Lotus 1-2-3 program at issue here, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass. 1990), and, the second, a Memorandum issued in this case, *Lotus Dev. Corp. v. Borland Int'l Inc., 788* F.Supp. 78 (D.Mass. 1992) (Memorandum and Order of March 20).

In the Memorandum and Order of March 20, I concluded that Lotus had failed to frame adequately its contentions with respect to the infringement of elements of its user interface less than the whole interface. In its renewed motion for summary judgment, Lotus asserts specifically that Borland has copied expressive elements of the 1-2-3 interface, including "menu commands," "menu structure," "long prompts," and "keystroke sequences." Although Lotus continues to argue that its entire user interface beyond dispute was copied, I adhere to the view that on the present record a reasonable jury could find that Borland copied less than the whole 1-2-3 user interface. Nevertheless, based upon the parties' most recent submissions, I conclude that, beyond *genuine* dispute, Fed.R.Civ.P. 56. Borland copied parts of the 1-2-3 user interface. For the reasons stated herein, I deny Borland's motion for summary judgment and grant, in part, Lotus' motion for summary judgment.

---

* Lotus argues that Borland has not filed a renewed motion for summary judgment. There is no docket notation of Borland's renewed motion, nor has such a motion been physically located by the court. However, several of Borland's submissions are submitted in support of "Borland's Renewed Motion for Summary Judgment." In this circumstance, and because I conclude that I must in any event deny such a motion, I will consider Borland's "motion" on its merits.

I. DEFINITIONS AND PREMISES

The Memorandum of March 20 presented for possible use in this case the following form of jury interrogatory concerning the extent to which Borland copied the Lotus 1-2-3 user interface in creating its Quattro programs:

Question 1

(a) Do you find that the Quattro Pro user interface as a whole was copied from the Lotus 1-2-3 user interface as a whole?

____YES ____NO

(b) Do you find that the part of the Quattro Pro user interface called the "emulation interface" (also called the "1-2-3 compatible interface") as a whole was copied from the Lotus 1-2-3 user interface as a whole?

____YES ____NO

(c) If NO, do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 user interface were copied into some part of the Quattro Pro "emulation interface" (also called the "1-2-3 compatible interface")?

(1) The menu commands	____YES	____NO
(2) The menu structure	____YES	____NO
(3) The command sequence	____YES	____NO
(4) The long prompts	____YES	____NO
(5) The macro facility	____YES	____NO

The Memorandum of March 20 noted that it was not clear that Lotus was making a claim of the sort addressed in part (a) of proposed Question 1 and that ambiguity remained regarding the meaning of the terms used in parts (c)(1)–(5). Lotus has in its recent submissions clarified its contentions.

First, Lotus acknowledges that the "native" modes of the Quattro programs have user interfaces that differ from that of 1-2-3. Thus, the question posed in part (a) of Question 1 is not in genuine dispute.

Second, Lotus has defined, as it uses them, the terms "menu commands," "menu structure," "keystrokes," "keystroke sequences," "long prompts," and "macro language."

In general, except for some blending of argument with definitions, the parties appear not to be in dispute about the meaning of these and other terms defined below. The definitions that I use in this Memorandum, for the purpose of explaining and analyzing the contentions of the parties, are consistent with the submissions of both Lotus and Borland, as I understand them.

"Command" refers to an abbreviated description of a direction that a user of a software program (whether Lotus 1-2-3, Borland's Quattro Pro, or another program) may invoke to cause some operation to be performed.

"Menu" refers to a display on the computer monitor of a limited number of commands available to the user at a given moment.

"Menu command" refers to a command that appears in a menu. In Lotus 1-2-3, a menu command is ordinarily a single English-language word. In rare instances, it is instead a representation of an English-language pronunciation (such as "Xtract"). Menu commands are displayed on the computer monitor by the 1-2-3 program in a succession of menus. The menus communicate to the user, in sequence, the spreadsheet operations available to the user.

"Command structure" refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include "menu command structure," "menu hierarchy," and "menu command hierarchy.") In Lotus 1-2-3, menu commands are organized so that less than a dozen related menu commands are displayed at any given moment. This display communicates to the user the spreadsheet operations immediately available. Each menu of less than a dozen commands is linked to preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a single main (root/trunk) menu to form a "menu tree."

"Keystroke sequence" refers to a sequence of keystroke entries that a user may invoke. Keystroke sequences may be

generated as one navigates the menu command hierarchy performing sequential spreadsheet operations.

"Long prompt" refers to a displayed multi-word English-language description of a "highlighted" menu command. A "highlighted" menu command appears on the computer monitor as a block of inverse video—that is, on a monochrome monitor with a black background on which characters are lit, a highlighted word appears as black letters within a lit block.

"Macro language" refers to a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence. Thus, a user may invoke the short keystroke sequence (a "macro") as a substitute for the longer keystroke sequence. In stating this definition, I omit a sophisticated programming capability available in 1-2-3 through its macro language feature that Lotus, as I understand its submissions, does not contend is involved in its claim of infringement in this action.

Having stated the definitions of the components of the user interface that I will use in this Memorandum, I now state additional points that I conclude are not in dispute about the relations among these definitions and associated matters.

The keystroke sequences and macro language have functionality. Typing ("inputting," in jargon) the first character of a command word invokes the command and causes the operation associated with the command word to be performed. (In many instances, a submenu associated with the command word is displayed.) The menu command hierarchy is a fundamental part of the functionality of keystroke sequences and the macro language. For example, the keystroke sequence "/RFC" directs the computer to format a range of numbers to appear as currency values because the character "/" initiates a command sequence, the character "R" implements the "Range" command, the character "F" implements the "Format" command, and the character "C" implements the "Currency" command.

It may be necessary to enter additional information to invoke a spreadsheet operation fully. For example, in order to implement an operation formatting a range of numbers as currency values, it is necessary to delimit the range of spreadsheet cells to be formatted. In addition, it is necessary to specify the number of decimal places to be displayed. Variables such as range of cells and number of decimal places, for which values must be input each time an operation is to be performed, are called "parameters."

The authors of Lotus 1-2-3 made certain predictions about the value of each of the relevant parameters likely to be input for use with certain spreadsheet operations. Those predictions have been incorporated into Lotus 1-2-3 as suggestions; a user failing to specify a value for a parameter where it is necessary to supply one accepts the suggestion of 1-2-3's authors by default. The "/RFC" command set, for example, has associated with it two "default parameters." The default for range is "current cell," and the default for number of decimal places is two (i.e., dollars and cents). The user who prefers a different format (for example, whole dollars rather than dollars and cents) may enter a different number, zero rather than two (the default) to so indicate. In similar fashion, the user may supply a range different than the default value.

The menu structure will not permit the command 'Currency" to be executed without first proceeding through the "Range" and "Format" commands. Indeed, inputting a "C" at a different point in the menu structure may cause a different command, such as "Copy," to execute.

The foregoing description identifies one way in which a menu command may be invoked—that is, by pressing the letter key corresponding to the English-language name of the command. A second way of invoking a menu command is to make use of the highlighting around a menu command. The user may use arrow keys on the keyboard to move the block of highlighting ("cursor") to an adjacent menu command. Depressing the "Enter" key (or "Return" or "<—" key, depending on the keyboard) invokes a highlighted command.

Because the macro language plays such a central role in the parties' contentions, and because it is an extraordinarily sophisticated element of Lotus 1-2-3, I recite some further examples of the use of the macro language.

A user may define a keystroke sequence with a macro by inputting the keystroke sequence in a spreadsheet cell and

assigning a macro label to that cell. For example, if a user enters the sequence "Hello~" (the tilde, "~," stands in for the "Enter" or "Return" or " <—i" key) in spreadsheet cell A1 and assigns the macro label "\H" to cell A1, then the user may cause the word "Hello" to appear in any other cell by invoking the abbreviated keystroke sequence "\H" instead of the longer sequence "H," "e," "l," "l," "o," "Enter." (The reverse slash, "\," signifies the "Alt" key, pressed and held in place while another key is pressed. Thus, the keystroke sequence "\H" consists of pressing the "h" key while the "Alt" key is simultaneously depressed.)

The capability of the program to enter, for example, the word "Hello" in any spreadsheet cell when the "\H" keystroke sequence is entered is functional. It is not protectable by copyright, 17 U.S.C. § 102(b), and it is not the subject of this case.

At a more sophisticated level, a user may construct a macro that implements menu commands. For example, a user may input the keystroke sequence "/RFC" in cell A1 and attach a macro label such as "\C" to that cell. This defines "\C" in the macro language subject to later redefinition—as "/RFC~." Invoking the "\C" keystroke sequence in any cell will cause the spreadsheet to format that cell to display numeric values as currency values. The ability of the computer to format the cell for currency values is functional. It is not copyrightable, *id.*, and it is not the subject of this case. The spreadsheet program is instructed to format the cell by the keystroke sequence '/," "R," "F," "C," "Enter," "Enter." That sequence invokes the menu commands "Range," "Format," and "Currency" (and accepts, by operation of the "Enter" key twice, two default parameters). Thus, the menu commands are an important part of the functionality of the macros. Lotus contends that the menu commands and the command structure are copyrightable expressive elements of the 1-2-3 user interface and that they are copyrightable expressive aspects of the macros. Borland contends that the macros, in their entirety, are an uncopyrightable system. These competing contentions, as well as others, are addressed below.

II. COPYING

In Borland's Mem. in Supp. of Renewed Mot. for Summ. J. (Docket No. 168), Borland states the following:

> Furthermore, it is undisputed in the record that Borland did not copy the 1-2-3 menu command hierarchy directly from any 1-2-3 version, including Release 2.01. Rather Borland employees reviewed books about 1-2-3, Release 2.01, written by third-parties, which books contain schematic or menu-tree type representations of the 1-2-3 menu command hierarchy. Borland used these third-party menu trees to construct 123-compatible menu hierarchies in their own products.
>
> But Borland employees did not copy those menu trees, even from the third-party books. Rather, they viewed the menu trees and implemented into their own products the relationship of functions depicted in those menu trees.

Id. at 15 (footnote omitted). Borland contends that, on these facts, indisputably Borland did not "copy." Borland's contention, however, is based on its idiosyncratic use of the word "copy," and is fundamentally wrong. Instead, Borland's admissions establish beyond dispute that Borland did copy the menu commands and command structure of Lotus 1-2-3.

Borland argues that the menu command hierarchy is a "set of functional relationships" that is nowhere displayed in the 1-2-3 user interface. Thus, Borland argues, it did not and could not have "copied" the menu command structure. That argument simply fails. It is an argument about a fact—copying or not. Its premise bears instead not on the fact-copying or not—but upon the legal issue of copyrightability. That is, the premise of Borland's argument is that the menu command structure must be fixed in a tangible medium to be copyrightable. *See* 17 U.S.C. §§ 101-102. The admitted fact that the Quattro programs duplicate the set of "functional relationships" of Lotus 1-2-3 and were designed to do so is conclusive against Borland on the issue of copying that set of functional relationships. Thus, Borland has admitted that it

intentionally incorporated into its user interface the 1-2-3 menu commands and menu command hierarchy.

Moreover, I reject Borland's tangible-medium argument as applied to this case. The argument would be relevant, if at all, to copyrightability rather than copying, but because Borland has made the argument as if it had a bearing on copying, I will digress briefly to address it here. To be the subject of copyright protection, an expression must be fixed in a tangible medium. *Id.* § 102. All that is required in this regard is that the expression be embodied in a copy "by or under the authority of the author" in a form "sufficiently stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." *Id.* § 101. The output of a computer program, at least insofar as it is typical of the program, predictable from it, and directed by the operation of the program, satisfies these requirements. The menu command hierarchy is part of the 1-2-3 program's output, is directed by the program, is identical each time the program runs, and may be perceived (and, as in this case, duplicated). It is irrelevant that the hierarchy cannot be perceived in its entirety at one moment (for example, in one screen display), just as it is irrelevant that the plot of a novel cannot be perceived from viewing one page. The menu command hierarchy is copyrightable subject matter.

Borland's argument that it copied from third-party sources rather than Lotus is equally without merit. Borland argues this point on the strength of the court's suggestion that Borland might prove that it did not copy Lotus' interface by demonstrating that instead it copied someone else's. If Borland had copied a third party's independently created menu command hierarchy that "happened" to duplicate Lotus' interface, Borland might be excused (at least from liability to Lotus, though not perhaps to the third party). Here, however, Borland has admitted copying from sources that, with or without permission from Lotus, copied from Lotus. Borland was aware that those sources copied from Lotus—Borland admits verifying the accuracy of the menu structure it generated from third-party sources by comparing the Quattro programs to 1-2-3. The fact that Borland used third-party sources as a means of

copying the Lotus 1-2-3 menu command hierarchy in no way excuses Borland's deliberate imitation of the Lotus menu structure.

Based on the foregoing, I conclude that, beyond genuine dispute, Borland copied the menu commands and menu command structure of Lotus 1-2-3. Moreover, Borland admits to copying the functionality of the keystroke sequences and macro language. By its own assertions, Borland's reason for copying the menu command structure was to obtain the benefit of its functionality. It follows that in fact Borland has admitted copying aspects of the keystroke command sequences and macro language that Lotus contends are expressive and copyrightable, although Borland chooses to describe what it did in another way.

Lotus has identified at this stage one additional element of its user interface that it claims Borland copied: the long prompts. It is clear from the record that the long prompts appearing in the Quattro programs differ in many instances from the long prompts in 1-2-3. On the other hand, in many other instances there is little or no difference. There is evidence that a Borland employee wrote the Quattro long prompts and did not "copy" from 1-2-3; however, she admits to looking at 1-2-3 to ensure that she correctly understood the Lotus commands. A reasonable jury could find on the basis of this evidence either way—that Borland did or that Borland did not copy the long prompts of 1-2-3. Therefore, I conclude that whether the long prompts were copied is a question for the jury. Moreover, because of this conclusion, it is apparent that a reasonable jury could find that Borland did not copy the 1-2-3 interface as a whole.

III. COPYRIGHTABILITY

Having concluded that Borland copied the menu commands and menu command hierarchy as well as the keystroke sequences and macro language, I now proceed to determine whether those aspects of the 1-2-3 user interface, taken together, are copyrightable.

A. Potential Fact Questions

I stated earlier my tentative conclusion that the application of the copyrightability standard is for the court and not a jury. *Borland*, 788 F.Supp. at 96. I invited counsel to respond to that tentative conclusion.

Lotus has fully endorsed the conclusion that copyrightability issues, at least in this case, are for the court. Borland, however, contends (in the alternative to Borland's own motion for summary judgment) that questions of copyrightability are for a jury. In response to the court's invitation, Borland proposes the following questions to be answered by a jury:

> 1. Does the Lotus 1-2-3 menu command hierarchy comprise a system, procedure or method of operation?
>
> 2. Is the 1-2-3 menu command hierarchy designed and used as a system for performing tasks using a spreadsheet program?
>
> 3. Are the command words of 1-2-3 and their order an inseparable part of a system for performing spreadsheet tasks?
>
> 4. Does the 1-2-3 menu command hierarchy enable a person to map out and execute a procedure for performing a particular spreadsheet task?
>
> 5. Is the 1-2-3 menu command hierarchy a means for issuing commands to the computer program to perform spreadsheet tasks?
>
> 6. Does the 1-2-3 menu command hierarchy provide a procedure or method of operating a spreadsheet program?
>
> 7. Was the 1-2-3 menu command hierarchy designed and arranged using functional rules or principles?
>
> 8. Was the 1-2-3 menu command hierarchy designed and arranged to maximize its efficiency and usefulness?
>
> 9. Is the 1-2-3 menu command hierarchy fundamental to a user's ability to execute macros written using 1-2-3?

> 10. Do the command words of 1-2-3 convey information to the user other than the choices of functions that are available?
>
> 11. Does the 1-2-3 menu command hierarchy explain to the user how to use the 1-2-3 program to perform spreadsheet tasks?

Liability Questions for the Trier of Fact (Docket No. 95) at 2-5. Under my rulings, it is irrelevant that the 1-2-3 interface includes functional elements or "comprises a system" so long as it also includes separable expressive elements. Thus, five of Borland's first six questions are irrelevant. Question 3, though relevant, is a mixed law-fact question that will not be asked of a jury for reasons previously stated. *Borland*, 788 F.Supp. at 94-96.

The seventh and eighth questions are more problematic. Those questions may have a bearing on whether the expressive elements of 1-2-3 are in fact separable from the functional aspects of the interface. As stated in the Memorandum of March 20, "[i]f the menu commands or menu command structure were dictated solely by functional concerns, then those elements may not be copyrightable." *Borland*, 788 F.Supp. at 97 (citing *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir.1987)). Lotus argues that *Brandir* is not "strictly applicable" because computer programs are classified as "literary works" and not "pictorial, graphic, and sculptural works" by the Copyright Act. The law draws heavily on analogy, however, and computer programs, whatever their formal classification, like pictorial, graphic, and sculptural works, are useful articles. Moreover, though insisting that the burden on this question falls on Borland, Lotus acknowledges that any elements of its program that were functionally dictated are not copyrightable. Pl.'s Mem. of L. in Supp. of Renewed Mot. for Summ. J. on Infringement (Docket No. 171) at 34-35 n. 47.

Nevertheless, I conclude that no reasonable jury could find that the menu command hierarchy was limited to one or even several alternate designs at the time it was created. On the

basis of the evidence before me, a factfinder could conclude that some—but only some—subelements of the menu command hierarchy were functionally dictated. For these reasons, it is my tentative conclusion that there may be genuine fact disputes regarding the subject matter of Borland's seventh and eighth questions. Questions seven and eight, as formulated, however, are "evidentiary" rather than decisive or "ultimate" issues appropriate for use on a verdict form.

It remains to be determined whether an "ultimate question" on the subject matter of the seventh and eighth of Borland's suggestions can be formulated as an adjudicative fact question of the type that is normally the province of a jury and does not pose the substantial risks described in my earlier Memorandum, *see Borland*, 788 F.Supp. at 94-96.

Question nine assumes a premise that is fundamentally in error—an error that has been systemic in Borland's arguments throughout the course of this litigation. The problem, which may be described as a "chicken and egg" problem, is addressed in greater detail in Section C, below. At this point, I simply state my view that question nine should not be asked of a jury.

The answers to the last two questions are irrelevant. A "no" answer to question ten does not preclude a determination that the command words are expressive even though the expression may be limited—i.e., the words communicate the functions to which they are assigned. Similarly, the answer to question eleven is immaterial.

B. Legal Issues on Copyrightability

In the Memorandum and Order of March 20, 1992, I concluded that, "absent further guidance from higher authority before the date of trial," *Borland*, 788 F.Supp. at 89, I would apply the following standard for deciding copyrightability issues:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

> that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing betwen the idea, system, process, procedure, or method and its expression.
>
> * * *
>
> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.
>
> THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

Id. at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphasis omitted).

The Second Circuit has recently issued an opinion that bears significantly on issues of copyrightability and substantial similarity. In *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 1992 WL 139364, No. 91-7893, 1992 U.S.App. LEXIS 14305 (2d Cir. June 22 1992), the court announced an "Abstraction-Filtration-Comparison" test for determining "substantial similarity." The first two steps, "abstraction" and "filtration," are designed to define the idea of the program and to eliminate it (as well as other noncopyrightable subject matter) from further consideration. The test "serves 'the purpose of defining the scope of plaintiff's copyright.' " *Id.* (quoting *Brown Bag Software v. Symantec Corp.*, 960 F.2d

1465, 1476 (9th Cir.1992)). Thus, the first two steps of the Second Circuit's "substantial similarity" test concern what other courts and commentators have called "copyrightability." Only the third step, "comparison," addresses similarities between the copyrighted work and the allegedly infringing work.

To what extent does the Second Circuit's "Abstraction-Filtration-Comparison" test differ, either substantively or in methodology, from the combination of copyrightability and substantial similarity tests tentatively adopted for this case in my Memorandum and Order of March 20?

I conclude that the standard for determining copyrightability stated in my Memorandum and Order is compatible with the abstraction-filtration portion of the Second Circuit's test. The Second Circuit founded its abstraction step on the opinions of Judge Learned Hand that were also the foundation of the first step of the copyrightability test stated in my Memorandum and Order. The second step of that copyrightability test parallels the Second Circuit's "filtration" step.

The third step of the Second Circuit test, "comparison," serves two functions. The first concerns the issue addressed in the third step of the "copyrightability" test I have tentatively adopted for this case— whether the expressive elements of the allegedly copyrightable work are a substantial part of it. I conclude that in this respect the two tests are compatible substantively though different in methodology. The other function that the Second Circuit's "comparison" step serves is emphasized in the term used to identify it—"comparison." The comparison is between the relevant portions of the allegedly infringing work and the expressive elements of the allegedly copyrightable work to ascertain whether any part of the allegedly infringing work is similar to expressive elements of the allegedly copyrightable work that are a substantial part of the allegedly copyrightable work (i.e., whether there is substantial similarity in the mixed law-fact sense). I conclude, again, that in relation to this comparison, the Second Circuit's test and the combination of the "copyrightability" and "substantial similarity" tests I have adopted tentatively are compatible substantively, though different in methodology.

Borland argues that the decision and reasoning in *Computer Associates* are contrary to the *Paperback* decision, and, as well, to my Memorandum and Order of March 20. In one respect only, however, did the Second Circuit explicitly so indicate. It criticized an incentive-based reason stated at one point in the *Paperback* opinion. The criticized argument, however, was by no means essential to the outcome in *Paperback*, and acceptance or rejection of that argument is not likely to affect the outcome in this case. In other respects, my reasoning in *Paperback* and in my Memorandum and Order of March 20, 1992 in this case was substantively consistent with the opinion of *Computer Associates*, as I understand that opinion, though, as I have explained above, different in methodology.

A particular example of significance to this case concerns Borland's argument that the Second Circuit's treatment of "compatibility" militates against copyrightability of the 1-2-3 interface. Borland extracts from the Second Circuit's opinion a determination that aspects of computer software cannot be subject to copyright if they are greatly circumscribed by the hardware or software with which they are designed to interact. That proposition, even if accepted as Borland has stated it, does not apply to 1-2-3. Borland's argument to the contrary must be rejected for reasons that I explain in Part C, immediately below.

C. Which Came First?

A familiar childhood riddle asks: Which came first—the chicken or the egg? As folk riddles often do (and lawyer's questions on cross-examination sometimes do), this riddle strongly suggests that any answer but one of two identified options is out of bounds. Perhaps from another perspective, however, one may recognize that the chicken and egg are one type of organism in different stages of a life cycle. Over a long span of time the form of that life cycle has evolved from an earlier form in which the chicken and the egg were not so distinct as they now appear to be. History, science, and

philosophy may provide another answer, or many others, different from the two suggested in the riddle.

Borland's brief poses a conundrum for this case in a form analogous to the chicken-and-egg riddle. Borland asserts that the 1-2-3 interface is not copyrightable because the menu command hierarchy "was dictated by the nature of the user macros with which it was designed to interact." Supp.Mem.Re: Additional Authority (attached to Docket No. 189) at 9 (quoting *Computer Assocs. Int'l v. Altai, Inc.*, 1992 WL 139364, at *24, No. 91-7893, slip op. at 50, 1992 U.S.App. Lexis 14305, at *67 (2d Cir. June 22, 1992)). An implicit premise behind this argument is that the menu command hierarchy was designed to fit the macros. The subtle suggestion is that the macros came first—that they were preexisting. Necessarily implicit in Borland's argument is the assertion that neither the final version of the menu command hierarchy nor any substantial part of it was preexisting when the macro language was created.

The thrust of the discussion of "compatibility" in *Computer Associates*, however, relies upon *proof* that what the program was designed to fit was already in existence before the program was designed to fit it. Thus, a program designed to fit hardware specifications cannot be protected by copyright unless the program contains expressive elements not substantially dictated by the hardware. Similarly, a program designed to interact with preexisting software, such as the operating system at issue in *Computer Associates* is not entitled to protection to the extent that it is constrained by the need for compatibility with the preexisting software. Thus, the rule makes sense if the premise of a preexisting functional limitation is valid.

In this case, however, there is a very simple answer to the question "Which came first?" The Lotus 1-2-3 interface—or at least a version of it—was written first. All user macros derive from it. Thus, Borland is simply wrong factually to argue that the 1-2-3 interface was constrained by the macros. On this issue, there is no genuine dispute of fact.

I assume in Borland's favor that, like the chicken and the egg, the macro language (as opposed to macros defined by

users using the macro language, which necessarily came later) evolved simultaneously with the menu commands that delimit it. Nevertheless, it is beyond dispute that the macro language did not evolve *first*.

Borland has argued, also, that the need to ensure that the menu commands in any given menu begin with a different letter is a functional constraint. Even that argument fails, however, because of the availability of other symbolic tokens; e.g., A, B, C, . . . or 1, 2, 3, . . .

It is no doubt true that the macros have functional significance. Moreover, as this court found in *Paperback*, the menu "system" is a fundamental part of the functionality of the macro language and the macros. As the Second Circuit has recognized, however, the fact that a form of expression takes on functional character does not remove it from the protection of copyright. *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1147 (2d Cir.1987); *see also Paperback*, 740 F.Supp. at 58.

Lotus used the slash character ("/") to initiate a command sequence; the character "R," to specify the command "Range"; the character "F," to specify the command "Format"; and the character "C," to specify the command "Currency." A user inputting the characters "/," "R," "F," and "C" in sequence (and supplying necessary additional parameters) will cause the computer to perform the operations associated with the specified commands.

If the keystroke sequence "/RFC—" were stored in a spreadsheet cell that had been assigned the macro "\C," then a user inputting the "\C" keystroke sequence would cause the computer to perform the same operations as would be invoked by the six-keystroke sequence "/," "R," "F," "C," "Enter," "Enter." Because the "\C" keystroke sequence is invoked by pressing the "c" key with the "Alt" key depressed as one would use a "Shift" key, it involves two keys but perhaps only one "keystroke." Thus, depending upon one's definitions, a saving of either five or six keystrokes would result.

Had Lotus preferred, it could have chosen, for example, the command "Scope" instead of "Range," "Appearance" instead of "Format," and "Money" instead of "Currency." Then the

user would invoke the sequence "/," "S," "A," "M" to establish a "money" appearance ($xxx.xx) for a scope of cells. In that case, the parallel to Borland's argument would be that the existence of "/SAM~~" macros in users' files (spreadsheet files with cells defining macros may be saved for future use) "dictated" the use of the "Scope," "Appearance," and "Money" commands in the menu command hierarchy. Obviously, the argument is without merit—"Range," "Format," and "Currency" are demonstrably acceptable choices. The fact that Lotus chose one command set for its first version may have made it "necessary" that future versions adopt the same command set; however, the initial choice of the command set was a free choice.

It bears repeating here that Lotus (though not Borland) was entitled to incorporate in version 2.01 the menu command hierarchy that it employed in earlier versions of 1-2-3. Thus, Borland's contention that the macros preceded and dictated the menu command hierarchy of version 2.01 fails because that fact is irrelevant. This is so because Lotus' exercise of creative expression that is at issue in this case (the menu command structures of versions 1A and 2.0) manifestly preceded the users' macros that incorporate the 1-2-3 command hierarchy.

It may be argued that the macros, by themselves, are functionally dictated by the 1-2-3 menu command hierarchy and so are not copyrightable independently of the copyrightability of the menu command hierarchy. Moreover, even if some macros are copyrightable, it may be that the owner of any copyright in a macro is the user who authored the macro and not Lotus. I am not faced with those questions in this case. I conclude only that Borland's argument rests on a false proposition. Thus, to decide this case I need not and do not decide whether Borland is prohibited from reading and interpreting the macros that have been created by users of 1-2-3. Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland's

command hierarchy, that would have presented a different case from the one now before me.

Borland did not obtain the right to expressive aspects of Lotus' command hierarchy merely because—if it be the case—the 1-2-3 program revolutionized the spreadsheet market. The menu command hierarchy has a functional aspect when incorporated into the keystroke sequences and macros. That functional aspect is separable from the expressive aspect that preceded it. Borland cannot obtain the right to use the macros and keystroke sequences just because the only means of doing so is by infringing expressive features of the Lotus 1-2-3 macro language and keystroke sequences.

The error of Borland's argument may be demonstrated by a simple hypothetical. If an uncopyrighted movie is made from a copyrighted novel (under the authority of an appropriate license), the public is free to copy all aspects of expression *unique to the movie*; however, the novel does not enter the public domain. Similarly, if the macros have uncopyrightable functional aspects, Borland does not infringe the copyright in Lotus 1-2-3 version 1A by duplicating the functionality in any way that does not copy the expressive elements of Lotus 1-2-3 version 1A, but it must not infringe upon expressive elements.

Borland's argument that *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), establishes a different rule that controls this case is rejected. This is not a case like *Baker v. Selden* in which the system depends on the use of the copyrighted matter. Borland has, in fact, designed a system (Quattro Pro's native mode), using macros and keystroke sequences and using an alternate command hierarchy, that is fully functional.

D. More on *Computer Associates v. Altai*

Some further points regarding the Second Circuit's opinion bear upon its applicability to issues in the present case.

1. *Beyond Program Code to Nonliteral Expression.*

The reasoning underlying the Second Circuit's "Abstraction-Filtration-Comparison" test extends beyond program code to nonliteral expression. That is, although the issue in that case concerned program code and its structural aspects, the Second Circuit based its conclusions on reasons of broader application. In fact, the court explicitly approved determinations with respect to the copyrightability of certain nonliteral, noncode (nonstructural) aspects of the 1-2-3 spreadsheet in *Paperback*. *See Computer Assocs.*, 1992 WL 139364, at *16, No. 91-7893, slip op. at 34, 1992 U.S.App. LEXIS 14305, at *46. Thus, from statements made in the Second Circuit's opinion, albeit *obiter dicta*, I draw support for my conclusion that certain expressive elements of the 1-2-3 user interface may be protected by copyright.

2. *Criticism of Whelan.*

Borland's argument that the Second Circuit, with one stroke, knocked out the "conceptual underpinnings" of *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir.1986), *cert. denied*, 479 U.S. 1031, 107 S.Ct. 877, 93 L.Ed.2d 831 (1987) and of *Paperback* cannot be sustained. First, *Whelan* remains good authority in the Third Circuit and may provide guidance for this court in the same way as the Second Circuit's opinion in *Computer Associates*. That is, for courts in the First Circuit, each of these decisions from other federal circuits is instructive and neither is controlling beyond the persuasive force of its reasoning. Second, and more to the point, the Second Circuit criticized the *Whelan* decision for reasons that in large part do not apply to the rulings I made in *Paperback* or to the rulings I have made and now make in this case.

The Second Circuit was critical of *Whelan* on two principal grounds—that it failed to recognize that programs may have more than one idea, and that it mistakenly asserted that all that is not idea is expression. To resolve those problems, the Second Circuit formulated its "Abstraction-Filtration-Comparison" test.

In both these respects, the rulings I have made in the present case are compatible substantively (though not entirely in methodology) with the Second Circuit's views rather than those of the Third Circuit, where those two circuit's differ. For example, as indicated in the *Paperback* opinion, I recognize that the 1-2-3 user interface can be described as incorporating more than one idea. Thus, the *Paperback* opinion identifies the idea of an electronic spreadsheet, the idea of having a readily available method of invoking the menu at command system (which I concluded merged with its expression as the "/" command), and the idea of the two-line moving cursor menu, among others.

3. *Baker Not Alone Controlling.*

In its most recent submissions, Borland once again insists that 17 U.S.C. § 102(b) and *Baker v. Selden* control this case, now asserting that the Second Circuit decision supports Borland's argument. Once again, Borland overstates the point. Borland is correct in observing that the Second Circuit treats *Baker* as the "starting point" in analyses of utilitarian works, including computer programs. That view is compatible with my earlier Memorandum and Order. *See Borland*, 788 F.Supp. at 92-93. Borland fails to note, however, that the Second Circuit went on to state:

> While *Baker v. Selden* provides a sound analytical foundation, it offers scant guidance on how to separate idea or process from expression, and moreover, on how to further distinguish protectable expression from that expression which "must necessarily be used as incident to" the work's underlying concept.

Computer Assocs., 1992 WL 139364, at *11, No. 91-7893, slip op. at 24, 1992 U.S.App. LEXIS 14305, at *32. The Second Circuit's decision cannot fairly be characterized as holding that *Baker* controlled the outcome in *Computer Associates*. The Second Circuit was sensitive not only to its duty of fidelity to precedent, but as well to its duty of fidelity to congressional mandates that came into existence long after *Baker* was decided.

4. *"Substantial Similarity" and "Copyrightability. "*

The Second Circuit referred to its test as one of "substantial similarity" and did not use the term "copyrightability" for any part of the test. I do not understand this difference of terminology to have substantive implications, however, and more especially, not for this case. Nor do I understand the Second Circuit's three-step test to be meant as a rigid barrier to alternate methods of analysis and decision. The court "advised" rather than mandated that the courts within the Second Circuit adhere to the three-step test announced. Moreover, a court in the First Circuit must take account of some degree of dissonance between the First and Second Circuits regarding the methodology of infringement analyses. *Compare Computer Assocs., supra, with Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600 (1st Cir. 1988). Although *Concrete Machinery* outlines a methodology different from the three steps of the Second Circuit's test, I do not understand the First Circuit to have mandated the order of analysis it described. *See Borland*, 788 F.Supp. at 86. I adhere to the view stated in my Memorandum and Order of March 20 that, under existing precedents, prudential concerns about case management may have a bearing on the order in which a court proceeds with its analysis in a complex copyright infringement case. *Id.* Thus, I conclude that I may and should proceed to a copyrightability analysis at this time even though some potentially material factual questions are still unresolved—facts regarding the long prompts, and facts regarding the extent to which the menu commands and command hierarchy may have been dictated by functional considerations.

E. Applying the Standard

1. *The First Step.*

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"— for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

One may describe a number of conceptions of the 1-2-3 user interface. A non-exclusive list, commencing with the most abstract and moving toward the particular, includes:

(1) Lotus 1-2-3 is an electronic spreadsheet.

(2) It is a menu-driven electronic spreadsheet.

(3) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command.

(4) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.

(5) Finally, one may conceive of the interface as that precise set of menu commands selected by Lotus, arranged hierarchically precisely as they appear in 1-2-3. Under this conception, the interface comprises the menu of commands "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit," linked by operation of the command "Worksheet" to the menu of commands "Global," "Insert," "Delete," "Column, " "Erase," "Titles, " "Windows, " "Status," and "Page," etc. (The completion of this proposed statement of the "idea," listing all of the more than 400 commands for which "etc." stands, would require several dozen more lines of text.)

Borland argues that the appropriate conception of the "idea" of the 1-2-3 interface is the fifth option. If that were the case, of course, there would be no elements of expression in the menu commands and menu command hierarchy and therefore no copyrightable aspects in them. The premise of Borland's argument is that an "idea" of Lotus 1-2-3 version 2.01 is complete compatibility with earlier versions of 1-2-3, and more precisely with macros generated for use with earlier versions. Borland argues that the precise menu commands and menu structure are necessary to such functional compatibility. Thus, the argument goes, the entire interface of version 2.01 is a functional system or "idea" and is not copyrightable. This argument is essentially tautological. As applied to any case involving a useful article, an argument of this kind would always define the idea to incorporate all the specifics of the particular expression of that idea in the allegedly copyrightable work. Nothing would be copyrightable under this methodology of analysis. The argument is an attempt to win by definition without focusing at any time on any substantive issue concerning the separation of idea and expression.

To select, at the opposite extreme, the very abstract statement of the idea of 1-2-3 as "an electronic spreadsheet" would be to draw an inappropriately abstract boundary between idea and expression. Thus, I concur in a fundamental principle of the *Computer Associates* opinion and reject the contrary proposition in *Whelan*.

Arguably, my Opinion in the *Paperback* decision, where no sharper focus was essential to the outcome, is consistent with accepting a conception of the idea that falls between the second and third formulations above. *See Paperback*, 740 F.Supp. at 67 (electronic spreadsheet having "menu structure"). In any event, I now explicitly recognize that for decision of the issues now before me the selection of functional operations that the spreadsheet performs must be considered part of the idea of the program. Copyrightability depends on expression distinct from the selection of the set of spreadsheet operations that can be performed.

I conclude that an appropriate conception of the "idea" or "system" of the 1-2-3 interface is the fourth of the five alternative conceptions stated above.

2. *The Second Step.*

> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

Does the Lotus 1-2-3 user interface include identifiable elements of expression? For reasons stated below, I conclude that it does.

A very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

This may be easily demonstrated. Recall the ten commands that appear in Lotus' main menu: "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit." One can imagine an entirely plausible spreadsheet in which the "Worksheet" command has been named, quite naturally, "Spreadsheet." Of course, this might require changing the "System" command to avoid two commands abbreviated "S," perhaps to "DOS." The "Quit" command could be named "Exit" without any other modifications. The "Copy" command could be called "Clone," "Ditto," "Duplicate," "Imitate," "Mimic," "Replicate," and "Reproduce," among others (in some cases requiring modification of other com-

mands in the menu). Additional possibilities include "Output" for "Print," "Draw" or "Chart" for "Graph," "Figures" or "Information" for "Data," "Scope" for "Range," and "Transfer" or "Relocate" for "Move."

Just these potential modifications of the main menu yield over 250 combinations of commands in the main menu with ten distinct first letters. Changes in submenus increase the number of possible menu hierarchies *geometrically*. Since there are dozens of independent submenus, the number of possible menu hierarchies is extremely large.

Borland argues that "[t]o hold that an idea, plan, method or art described in a copyright is open to the public but that it can be used only by the employment of different words and phrases which mean the same thing, borders on the preposterous." Borland's Resp. to Pl.'s Renewed Mot. for Summ. J. (Docket No. 183) at 19 (quoting *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755, 64 S.Ct. 1265, 88 L.Ed. 1584 (1944)). This case, however, unlike *Crume*, is not a case in which the system "can be effected solely by the employment of words descriptive thereof." *Crume*, 140 F.2d at 184. Use of just the initial letters of command words (together with long prompts) or of other symbolic tokens would have been a sufficient alternate method of implementing the system. In this case, the command words chosen are not necessary to expression of the system nor are they necessarily incident thereto. *See Computer Assocs.*, 1992 WL 139364, at *13, No. 91-7893, slip op. at 30, 1992 U.S.App. LEXIS 14305, at *40.

Lotus argues that a large number of substantially different arrangements (hierarchies) could also have been effected. Looking again at just the main menu, is there any reason that the commands "Copy" and "Move," for example, could not have been arranged in the opposite order? Borland argues that the arrangement was necessary, citing evidence that Lotus arranged the menu commands in order of the expected frequency of use.

It is clear that certain command words have been grouped according to function; e g., there are eighteen commands that

affect the display mode of spreadsheet cells that are grouped together under the "Format" command. Thus, a jury could find that at least some aspects of the arrangement of command words, as opposed to the specific choice of command words, was guided by functional concerns.

This is, however, a disputable fact question that may affect only the scope of relief in this case. *See ABKCO Music, Inc. v. Harrisongs Music Ltd.*, 508 F Supp. 798 (S.D.N.Y.1981) (court awarded damages based upon contribution to success of infringing work of copyrighted material) (for subscquent history, *see* 944 F.2d 971 (2d Cir.1991)). I conclude that it cannot be genuinely disputed that a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations. There are sufficient non-functional aspects that at least hundreds and perhaps thousands of different expressions of the function were possible when Lotus chose the particular structure of menu commands incorporated into Lotus 1-2-3.

This may be demonstrated by examining more closely Borland's argument that the menus were arranged according to the predicted frequency of use of the commands. I assume the truth of Borland's assertion that Lotus predicted before marketing its spreadsheet that the "Copy" command would be used more often than the "Move" command. Nevertheless, that is merely a prediction of frequency of use. It did not require Lotus to list "Copy" before "Move." A user can type a "C" or an "M" with equal ease no matter which command is listed first. If a user prefers to invoke a command by first highlighting it and then typing "Enter," "Move" is only one keystroke from "Copy"; moreover, the same "right arrow" key that takes the cursor from "Worksheet" to "Copy" moves the cursor from "Copy" to "Move." Thus, the order in which commands are listed in a menu has very limited functional value.

In addition, a prediction of frequency of use depends upon who is the predicted user and what the predicted uses are. For example, a user may work on a spreadsheet without printing any of the work performed on that day. Yet, the user will ordi-

narily invoke the "Quit" command appearing at the end of the main menu at the end of each session. Nevertheless, the "Print" command appears before the "Quit" command on the menu.

Many users may rarely invoke the "System" command available in 1-2-3. That is, the need to invoke a disk operating system ("DOS") "shell" from within the spreadsheet program may be for most users of 1-2-3 a rare event. If the commands in the main menu of 1-2-3 are listed in order of predicted frequency of use, why does the "System" command "precede" the "Quit" command? In fact, what does it mean to say "precede"? "Q" neither precedes nor follows "S" on the keyboard. "System" precedes "Quit" as one moves from left to right within the main menu using the "right arrow" key. It is also true that "Quit" precedes "System" as one moves from right to left using the "left arrow" key. The 1-2-3 menus are circular (in jargon, they have a "wraparound" feature)—moving the cursor one step beyond the "end" of the menu causes the cursor to come to rest at the opposite "end." Thus, from the default cursor position on the "Worksheet" command, arguably "Quit" precedes "System."

For all these reasons, any ex ante prediction of frequency of use is itself of limited usefulness. It follows that the arrangement of menu commands according to predicted frequency of use is not a major functional limitation on the number of arrangements of menu commands.

The menu command hierarchy is an integral part of the functionality of the macros and of the keystroke sequences. Nevertheless, the fact that the macros and keystroke sequences incorporate the menu command hierarchy into their functionality does not remove the menu command hierarchy from the scope of copyright, if otherwise subject to copyright protection. Moreover, the macros and keystroke sequences are protected to the extent that it is necessary to infringe a copyright to use them. Of course, as I have stated above, it was *not* necessary to copy expressive aspects of the macro language and keystroke sequences to copy their function.

3. *The Third Step.*

> THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

The question posed by this element of the copyrightability test is whether the creativity involved in establishing the menu commands, menu command hierarchy, macro language, and keystroke sequences was more than trivial. No reasonable jury could find otherwise. As Borland has itself acknowledged, at least implicitly, Lotus 1-2-3 was a dramatic change and improvement over what was available on the market at the time Lotus was created. Although a large portion of that improvement relates to the functional aspects of Lotus 1-2-3, the features that I have now concluded are expressive also played a substantial role. Borland has maintained that those features are part of an uncopyrightable system (an argument I now reject for the reasons stated), but Borland has never argued that they were trivial, nor could it do so persuasively.

F. *Ashton-Tate v. Ross*

Borland argues that the Ninth Circuit's decision in *Ashton-Tate Corp. v. Ross*, 916 F.2d 516 (9th Cir.1990), *aff'g* 728 F.Supp. 597 (N.D.Cal.1989) militates in favor of a conclusion that the menu commands and command hierarchy are not copyrightable. The Ninth Circuit held that a list of menu commands was not copyrightable for reasons stated by the district court. The district court, in turn, held that a document bearing "a list of labels for user commands, many of which are common commands that were already available on other software programs" was not innovative or novel. *Ashton-Tate Corp. v. Ross*, 728 F.Supp. 597, 602 (N.D.Cal.1989). The relevance of that conclusion to this case is in some doubt in view of the fact that Lotus 1-2-3 is one of the programs on which

the commands were already available. There is no evidence in this case that the commands available in 1-2-3 were common commands at the time of Lotus' authorship.

The district court also held that there was nothing innovative in the order in which the commands were listed. *Id.* Borland has submitted in this case a copy of the list of commands at issue in *Ashton-Tate*. Borland's Mem. in Supp. of Cross-Mot. for Summ. J. (Docket No. 141) at 85-86. That list contains well under one hundred commands scrawled on one sheet of paper segregated into under one dozen functional groups. In form, detail, arrangement, and content, it bears almost no resemblance to the hundreds of menu commands arranged in Lotus' dozens of menus. The relevance of the court's conclusion to the case at bar extends, if at all, only to the copyrightability of subelements of the 1-2-3 menu hierarchy, and depends upon factual inquiries that remain to be made.

Finally, the court concluded that in the absence of any contribution to the user interface other than the command list (including, especially, any contribution to the program code), the contribution of the command list was a mere contribution of ideas. As Judge Learned Hand observed, however, and many others have agreed, decisions in which a line is drawn between idea and expression have an ad hoc character. They tend to be fact sensitive and case specific. The absence of other copyrightable contributions in *Ashton-Tate* may have caused the court to reach a determination in that case with respect to the copyrightability of menu commands that it might not have reached had the defendant contributed other copyrightable elements.

In the interest of completeness and candor, I note as well that courts in one circuit are not bound by the decisions of other circuits. I view my obligation as one of determining the law manifested in the Copyright Act as it would be interpreted and applied to this case by the Court of Appeals for the First Circuit and the Supreme Court should this case reach either or both of those courts.

IV. SUBSTANTIAL SIMILARITY (ILLICIT COPYING)

For the reasons stated in Part II, above, and supplemented briefly here, I cannot conclude that Borland has copied substantially the whole of the Lotus 1-2-3 user interface.

In *Paperback*, although witnesses called attention to "look and feel" as a descriptive metaphor, counsel for Lotus did not base their contentions on this metaphor, and the court in its decision did not rely upon it. 740 F.Supp. at 62-63. One reason the metaphor did not seem useful in that case is that the concept of "look and feel" relates more to substantial similarity (in the mixed law-fact sense) than to copyrightability. In *Paperback*, so much of the 1-2-3 user interface had been copied that it was not difficult to resolve questions of substantial similarity in the mixed law-fact sense. The difficult questions in that case centered only around copyrightability of the user interface.

Paperback involved an appropriation by the defendant of the entire "look and feel" of 1-2-3. In this case, Borland has appropriated, to a great extent, the "feel" of the 1-2-3 user interface and only to a lesser extent the "look" of 1-2-3. Indeed, Borland has designed an interface that in many respects looks substantially different from the 1-2-3 user interface. Borland's colors and pull-down menus are but two examples of the differences in "look" in the Quattro programs. Of course, the menu commands and the menu command hierarchy look the same in both programs.

The "feel," on the other hand, of the emulation modes of the Quattro programs depends in large part on the keystroke sequences one enters to perform spreadsheet operations. One enters the same keystroke sequence to perform the same spreadsheet operations in both 1-2-3 and Quattro Pro's emulation mode. They feel the same. Thus, an experienced user accustomed to the 1-2-3 interface needs to look seldom, if at all, to achieve the desired result in the emulation modes of the Quattro programs.

This conclusion may be expressed in a more straightforward way without use of any part of the "look and feel metaphor." Lotus is entitled to a judgment of infringement only if Borland appropriated copyrightable elements of 1-2-3 and those copyrightable elements, taken together, make relevant portions of Borland's program substantially similar to 1-2-3. The evaluation of substantial similarity (in the mixed law-fact sense) therefore depends upon determining what copyrightable elements of the Lotus 1-2-3 user interface Borland copied. Even if I assume, however, that Borland did not copy the long prompts, and that some aspects of the menu commands, menu command hierarchy, macro language, and keystroke sequences of 1-2-3 are not copyrightable, I conclude that no reasonable jury, applying the law, could find other than that the Quattro programs infringe 1-2-3. That is, a reasonable factfinder must conclude that the Quattro programs derive from illicit copying. The emulation interfaces are substantially similar in the mixed law-fact sense to the Lotus 1-2-3 user interface. (Returning to the metaphor, one may say that is why they "feel" the same.)

I am not able to determine on motion for summary judgment the precise scope of Borland's infringement. Genuinely disputable factual questions exist that may affect the scope of substantial similarity and therefore the nature and scope of the remedies for infringement in this case. Nevertheless, it is clear that Borland has infringed the 1-2-3 user interface, at least in substantial part.

Lotus argues that questions of substantial similarity in the mixed law-fact sense ought not to be put to a jury. I understand Borland to contend otherwise (though it does not explicitly address the point in its latest submissions, perhaps because the Memorandum and Order of March 20 may have left the impression that I would submit such questions to a jury if a genuine dispute of fact existed). The scope of issues to be put to the jury is significantly reduced by my ruling that the emulation modes of the Quattro programs have a core that is substantially similar to 1-2-3. Borland's request that I put questions of fact concerning substantial similarity in the

mixed law-fact sense to a jury other than the one I empanel to hear evidence regarding substantial similarity in the evidentiary sense is now moot.

Borland's Quattro programs infringe Lotus 1-2-3 because the extent of copying of copyrightable elements of 1-2-3 renders the Quattro programs substantially similar to 1-2-3. If the jury finds that Borland also copied other copyrightable elements, then the Quattro programs will, as a matter of law, be even more substantially similar to 1-2-3; that is, the scope of Borland's infringement will necessarily be broader. For example, if the jury finds that Borland copied the Lotus 1-2-3 long prompts, and if I conclude that the long prompts are copyrightable expressive elements of 1-2-3, those two determinations, without any further finding that Borland's long prompts are substantially similar to Lotus' long prompts in the mixed law-fact sense, support a conclusion that Borland's Quattro programs infringe the long prompts of 1-2-3. There would be no need to ask separately whether the copying of the long prompts would alone, or in combination, render the Quattro programs substantially similar to Lotus 1-2-3. Substantial similarity (in the mixed law-fact sense) is determined by comparing the copied copyrightable elements of the infringing work *all together* with the copyrighted work as a whole. I have already determined that this comparison, even if the long prompts were not copied, requires a determination of infringement.

In reaching these conclusions, I once again reject Borland's contention that it is entitled to place in evidence at trial all the elements *of Quattro Pro* that were not copied from Lotus. Despite Borland's otherwise detailed (if not precisely accurate) explication of the *Computer Associates* case, Borland has failed to acknowledge an express conclusion of the Second Circuit that is contrary to Borland's position:

> [I]n some cases, the defendant's program structure might contain protectable expression and/or other elements that are not found in the plaintiff's program. Since it is extraneous to the allegedly copied work, this material would have no bearing on any potential substantial sim-

> ilarity between the two programs. . . . Furthermore, by focusing the analysis on the infringing rather than on the infringed material, a court may mistakenly place too little emphasis on a quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protected expression.

Computer Assocs., 1992 WL 139364, at *22, No. 92-7893, slip op. at 47-48, 1992 U.S.App. LEXIS 14305, at *64-*65.

That Borland, in developing the Quattro programs, has added functional and expressive elements that do not exist in 1-2-3 is irrelevant in view of the fact that Borland copied virtually the whole menu command structure of 1-2-3 into its emulation interfaces. Borland's additions have caused some variation in the manner in which the elements taken from 1-2-3 are expressed in the Quattro programs. For example, in the main menu, the "View" command, a command not present in the main menu of 1-2-3, is interposed between the "System" and "Quit" commands. A decisionmaker in this case (whether judge or jury) must ignore the added expression *to the extent that it does not change the expression Borland copied from Lotus*. I conclude that no reasonable jury could find for Borland that Borland did not take the menu commands, menu command structure, macro language, and keystroke sequences substantially as they were.

V. BORLAND'S DEFENSES

Lotus has renewed its motion for summary judgment that Borland's affirmative defenses fail. Borland has raised three affirmative defenses in this action: waiver, laches, and estoppel. The issue of Borland's affirmative defenses was raised and fully briefed by the parties in conjunction with their previous motions for summary judgment. Because I denied both motions on other grounds by Memorandum and Order of March 20, there was no need to consider Borland's affirmative defenses at that time and I did not do so. Having concluded that Borland has infringed Lotus' copyright in 1-2-3,

I proceed to address Lotus' motion for summary judgment on Borland's alleged defenses.

A. Waiver

The parties agree that to succeed in its defense of waiver, Borland must demonstrate that Lotus voluntarily and intentionally relinquished its right to pursue Borland in this action. Borland's argument, however, is based upon *Borland's* "state of mind" and Lotus' alleged *concealment* of Lotus' "state of mind." Borland has not presented any evidence that Lotus manifested to either Borland or the world that it would not pursue Borland in this action. Borland's allegation that Lotus sought to play down the import of the *Paperback* litigation in its public relations campaigns by stating that *Paperback* involved "99% clones" does not amount to waiver, even if proved. Therefore, I grant Lotus' motion for summary judgment on the defense of waiver.

B. Laches

The parties agree that to succeed in its laches defense, Borland must demonstrate that Lotus delayed unreasonably in bringing this action and that as a result Borland suffered undue prejudice. Borland argues that it expended $18 million developing the Quattro programs, at least in part because, by not suing earlier, Lotus led Borland to believe that Lotus would not sue. Borland has admitted that it would have developed a spreadsheet program in any event at substantial cost. Nevertheless, Borland argues that the costs of releasing the entire spreadsheet constitute prejudice. This argument is plainly without merit.

Borland also argues that its decision to offer a 123-compatible interface is hard to undo and that it has expended large sums in advertising keystroke and macro compatibility of the Quattro programs with Lotus 1-2-3.

Without deciding at this time issues regarding alleged undue delay, I note that the question of prejudice may depend

upon determinations yet to be made regarding the appropriate remedy or remedies for the kind and scope of infringement that has occurred in this case. I take note also that precedents on laches derive from equity, and I may rule at trial that evidence admissible only in relation to equitable defenses must be received out of the presence of the jury, or in a later phase of trial. I deny Lotus' motion for summary judgment on the laches defense without prejudice to Lotus' renewing its arguments in support of the motion at a later date.

C. Estoppel

Estoppel, like laches, has its origins in equity. Without finally ruling on the matter at this time, I take note that it may be appropriate either to receive evidence bearing solely on estoppel (or only on estoppel and laches) out of the presence of the jury, or at a phase of trial after that in which issues relating to the scope of infringement are finally resolved.

Borland argues that Lotus' intentional concealment of its belief that the Quattro programs infringe 1-2-3 is a ground for estoppel. Borland's argument rests on a premise borrowed from an explicit statutory requirement in the law of patents, that Lotus owed a duty to Borland to notify Borland of its infringement. The law of copyright has not imported that requirement. Thus, Borland's arguments that Lotus concealed its intentions and that Borland did not and could not have known Lotus would sue are meritless. Lotus 1-2-3 is copyrighted. Borland copied copyrightable elements of 1-2-3 that constitute a substantial part of that program. Lotus has sued, and Borland is liable.

I reject as well Borland's argument that Lotus' delay in filing suit is a ground for estoppel. That argument is merely a repetition of the argument Borland raised in support of its laches defense.

Borland's identification of the alleged Lotus misrepresentation on which Borland relied to its detriment is less than pellucid. Borland refers to affirmative indications by Lotus that it would not sue, but fails to identify what indications Borland alleges, if any, other than comments made for public

relations purposes and reported in trade publications and position papers. Borland says, for example, that Lotus characterized the *Paperback* litigation as involving "99% clones." Borland argues that the Quattro programs are not clones of 1-2-3 and that it reasonably came to believe, therefore, that Lotus would not sue. If this is the best argument Borland can advance for justifiable reliance, the likelihood that it can present an issue on estoppel for a factfinder (judge, if this is treated as an issue in equity, and jury otherwise) seems very remote. Though I will not grant Lotus' motion for summary judgment at this time, the Order Regulating Trial in this case will require evidence bearing only on estoppel (or on estoppel and laches) to be proffered first outside the presence of the jury.

VI. CONCLUSION

It is my conclusion that, as a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3. Nevertheless, I conclude that a jury trial is essential before final disposition of this case because the scope of relief available will depend in part on whether the jury finds for Lotus on disputed factual contentions that the copying of separable expressive elements of the Lotus 1-2-3 user interface into the Quattro programs was greater than the minimum essential to constituting a substantial part of the Lotus 1-2-3 work.

In summary, Lotus' Motion for Summary Judgment is allowed in part and denied in part, as stated in the foregoing rulings. Borland's motion for summary judgment is denied.

Having reached this conclusion, I will state these rulings in the form of Determinations of Undisputed Facts and Con-

clusions of Law incorporated as a section of a draft Order Regulating Trial, to be distributed to counsel separately from this Memorandum and Order. An Order Regulating Trial will be entered at or after the Pre-Trial conference scheduled for September 23, 1992. Each party is invited to submit, by Memorandum filed on or before September 1, 1992, proposed modifications, deletions, and additions to the draft Order Regulating Trial. Responses may be filed on or before September 15, 1992.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

March 20, 1992

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, Mass., Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, Mass., Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, Cal., David L. Hayes, Fenwick & West, Gary L. Reback, Wilson, Sonsini, Goodrich & Rosati, Mitchell Zimmerman, Fenwick & West, Palo Alto, Cal., Peter Erich Gelhaar, Donnelly, Conroy & Gelhaar, Boston, Mass., *for defendant.*

MEMORANDUM AND ORDER

KEETON, *District Judge.*

In this civil action, the plaintiff, Lotus Development Corporation ("Lotus"), seeks damages and equitable relief for alleged infringement by defendant, Borland International, Inc. ("Borland"), of the Lotus copyright in its computer software

program, Lotus 1-2-3. This is the same copyright for the infringement of which Lotus has obtained relief under this court's decision in *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass. 1990) (holding that Paperback's VP Planner was an infringing software product).

Lotus contends that on undisputed facts Borland's Quattro and Quattro Pro (herein, collectively, the "Quattro programs") are likewise infringing. Borland responds that its products are materially different from both Lotus 1-2-3 and VP Planner and that the court should determine on undisputed facts that the Quattro programs do not infringe the Lotus copyright. Borland argues that this result follows under the rules and standards of law applied by this court in *Lotus v. Paperback*. In the alternative, Borland contends that this court should reconsider its rulings of law in the *Paperback* case and now hold that the elements of Lotus 1-2-3 it determined previously to be copyrightable are not copyrightable. Also, Borland contends that in any event it has valid defenses that Paperback did not have.

Pending before the court are cross-motions for summary judgment, fully briefed and with factual submissions complete except for confidential materials to be presented to the court pursuant to a stipulation and protective order.

For the reasons explained in this Memorandum, I conclude that neither party's motion is supported by the record now before the court.

The primary contention of Lotus that Borland copied the Lotus user interface as a whole fails because there is a genuine dispute of fact as to whether Borland did so. Borland's primary contention that it is entitled to a summary judgment on copyrightability fails on legal grounds for reasons explained in Part VI of the Memorandum.

Each party has advanced alternative contentions for summary judgment. Each of the submissions before me, however, fails to meet the movant's burden of identifying both a clearly stated legal theory and a clear statement of undisputed facts sufficient to demonstrate an entitlement to summary judgment. In these circumstances, I conclude that it is appropriate

to deny the cross-motions for summary judgment. Explicitly, however, I do not preclude the filing of another motion by either party if one can be presented consistently with the rulings stated and explained in this Memorandum.

I. *Redacted Submissions*

Redacted submissions (Docket Nos. 141 and 147 and certain affidavits, declarations, and exhibits) now before the court were filed pursuant to this court's practice (for the protection of parties, counsel, and court personnel from risks of inadvertent disclosure) not to allow protected materials to be filed with the Clerk before the judge (not any other person acting under a delegation of authority) has approved in writing the specific filing.

In this instance, I have concluded that the description of the protected materials, appearing in the redacted filings, is sufficient for me to determine that they are not material to any of the rulings stated and explained in this Memorandum. If counsel think otherwise, however, they may, with notice to the court and opposing counsel, bring the protected materials to the next scheduled conference to be presented to the court for examination during that conference. In light of this ruling, the parties' motions regarding the submission of confidential documents (Docket Nos. 140, 156, and 160) are dismissed as moot.

II. *Disposition of the Cross-Motions for Summary Judgment*

This court held in *Paperback* that the 1-2-3 user interface, taken as a whole, was copyrightable. That ruling was never appealed or vacated and continues to have the limited precedential force of a district court decision. The applicability of that precedent to this case is disputed, however, because Borland contends that its allegedly infringing products are materially different from VP Planner (the computer work involved in *Paperback*). VP Planner has been described, both in and

outside this court, as an imitation (or look-alike, or clone) of 1-2-3. The conclusion that the user interface of 1-2-3, as a whole, was copyrightable was outcome determinative as to a computer work like VP Planner because, without dispute, the 1-2-3 user interface as a whole was copied. Lotus has maintained, however, both before this court and elsewhere that there is a distinction between imitations of 1-2-3 and other products such as Microsoft Excel.

In the present case, unless other issues are dispositive, it will be necessary to determine on which side of an unmarked boundary, between imitations of 1-2-3 and products such as Excel, the Quattro programs fall. I cannot determine on the record before me that the "1-2-3 interface" (also called the "emulation interface") of the Quattro programs is indisputably an imitation of the 1-2-3 user interface. Thus, for Lotus to meet its burden of showing entitlement to a summary judgment of infringement it must (a) identify expressive elements in 1-2-3 that were indisputably copied in the Quattro programs, (b) establish that those expressive elements, either separately or together, are as a matter of law copyrightable, and (c) establish that the copied expressive elements of the Quattro programs' emulation interfaces are substantially similar to copyrightable elements of the 1-2-3 interface.

The conclusion that the user interface as a whole is copyrightable (which this court reached in *Paperback*) does not resolve the further questions that may now have to be resolved regarding the copying and copyrightability of individual parts or a sum of parts less than the user interface as a whole. The Lotus motion and supporting submission fail to focus precisely, even in the alternative, any claim of copyrightability and undisputed copying of something less than the user interface as a whole. In these circumstances, having concluded that the claim of copying the user interface as a whole has not been demonstrated, I must conclude also that Lotus has failed to demonstrate it is entitled to summary judgment.

Similarly, Borland has failed to demonstrate support for its contention that, even if it copied something from 1-2-3, nothing Borland copied was copyrightable. As far as I am now

able to determine, a factfinder may find on disputed evidence that Borland copied the 1-2-3 user interface as a whole. I have previously determined, and as explained in this Memorandum now adhere to the view, that the 1-2-3 user interface as a whole is copyrightable. Moreover, the present record suggests that a factfinder surely could find that Borland copied some expressive elements of the 1-2-3 user interface. The outcome of this case may depend on a more precise focus than Borland has presented as to what elements Borland copied and whether one, or more, or some set or sets of those elements are copyrightable.

One effective way of focusing contentions and considering whether they present genuine disputes of fact that are material under the court's resolution of contested issues of law is to consider how any fact questions might be framed for the factfinder's consideration. Especially is this so when a demand for jury trial has been filed. Questions to be submitted to the jury must at some point during full preparation for trial be framed precisely. The trial judge and trial lawyers may well commence this task early, to aid as well the focusing of arguments on motions for summary judgment. A lawyer who contends that a genuine dispute exists as to some material fact should be able to frame clearly a "written question[ ] susceptible of categorical or other brief answer," proposed for use in a "special verdict," Fed.R.Civ.P. 49(a). If, when challenged to do so, counsel cannot state precisely any proposed question of fact that is disputed and material, summary judgment is almost certainly appropriate. If, on the other hand, counsel can frame even a single disputed and material question of fact, summary judgment is inappropriate.

On the present submissions, I am not able to conclude that either party has met its burden of showing a basis for summary judgment in its favor. I therefore deny the motions at this time; however, I conclude that it is appropriate to allow the parties an immediate opportunity to focus their arguments more precisely, on the chance this may lead to disposition without the high costs of trial, and in any event will better focus issues for trial if a trial is necessary. Therefore, I deny

the motions now before me, but will allow each party, on the schedule stated below, to file a new, better focused motion if the party concludes it can surmount the serious obstacles identified in this Memorandum.

The claims and defenses in dispute in this case present issues of law and issues of fact that are independently complex. The relationships among the law and fact issues dramatically enhance complexity. A jury demand and the resulting need for clear and explicit explanations to jurors of the law they need to understand in order to identify and resolve material disputes of fact produce still more consequences of significance, and perhaps more complexity as well. Complexities and their consequences have significant implications for fundamental issues of fair adjudication. They present also some pragmatic issues of case management in both pretrial and trial proceedings. The Order entered at this time is aimed at creating a context in which the parties and their attorneys will have incentives to use their adversary presentations in ways that will better focus and illumine issues for the decisionmakers, judge and jury, and will save the parties, their counsel, and the public from needlessly excessive litigation costs. Suggestions of counsel in aid of this aim are invited and encouraged.

III. *Entanglement of Law and Fact as to Copyrightability and Substantial Similarity*

A. Basic Methods of Separating Law and Fact

In general, an issue of law is decided by a court—not by a jury in a jury trial and not by a trial judge as factfinder in a nonjury trial. Also, in general if reasonable persons can differ about how a fact question should be decided on the evidence received in the trial, the jury decides it in a jury trial, and the jury decision is final; in a nonjury trial, the judge as factfinder decides, and the decision is subject to review under a deferential standard. Fed. R.Civ.P. 52(a) (findings sustained unless "clearly erroneous").

These clear and straightforward rules sometimes require supplementation, however, when the issues of law and fact are "mixed." If a court can clearly separate law from fact, it may use either of two basic methods of submitting questions to a jury. One method asks the jury to answer a mixed law-fact question; the other, a strictly factual question.

Under the first method, the court submits a combined law-fact question to the jury. To use this method properly the court must first decide the issues of law so it can give the jury clear instructions on the law. The jury's answer, if they discharge their responsibility well and truly, decides only fact questions even though their answer is in form an answer to the mixed law-fact question.

This first option may be used either for a "special verdict" under Rule 49(a), or for a "general verdict accompanied by answers to interrogatories" under Rule 49(b) of the Federal Rules of Civil Procedure.

The second option open to the trial judge is to frame "special questions" for the jury under Rule 49(a), or "interrogatories" under Rule 49(b), that ask only fact questions, not mixed law-fact questions.

A charge that directs the jury to return only a general verdict cannot use the second method because the general verdict is necessarily a mixed law-fact finding (the type required under the first option). Thus, a verdict "for the plaintiff" or "for the defendant" does not provide any basis for a reasoned determination by the court that the verdict rests on findings of fact supported by evidence and not on a different view from that of the court about the legal elements of mixed law-fact determinations.

If only a "special verdict" under Rule 19(a) is to be returned, and the questions are well framed, the jury need not have any explanation of the legal rule or rules the court will apply to determine what judgment to enter in view of the jury's findings. The judge may choose to explain anyway, but giving the explanation is a matter of choice rather than necessity.

When two or more questions of fact must be submitted, the court may submit one or more questions by one of these meth-

ods and one or more by the other. Also, the court may use a mixture of general verdict and "interrogatories," Fed. R.Civ.P. 49(b).

When issues of law and fact are easily separable, the court, when it perceives a need to do so to avoid jury confusion or bias, may use a "special verdict" form under Rule 49(a) that strictly submits fact questions; also, the court may cleanly separate the hearings before the court on the legal issues from those before the jury on the fact questions. A key purpose of trying a case in this way is to reduce the risk that the jury will be improperly influenced by evidence and arguments not admissible in relation to the strictly factual issues submitted to them.

When law and fact are not easily separable—regardless of how deep the entanglement may be—it nevertheless remains exclusively the responsibility of the court to decide the legal questions.

Ordinarily, it is the responsibility of the factfinder (jury or, in a nonjury case, trial judge) to decide the fact questions, but exceptions exist, as will be noted in Parts IV, VI, and VII below.

B. Problems of Entanglement

In order to use a strictly factual special verdict form, a court must cleanly separate law from fact. Not all mixed questions of law and fact can be easily separated. Indeed, there may be instances in which law and fact are so deeply intertwined that, at least as a practical matter if not strictly in principle as well, total separation cannot be achieved. In copyright cases, is "substantial similarity" such an instance? Is "copy-rightability" such an instance?

1. *Substantial Similarity*

"Substantial similarity" is not consistently used with a single, settled meaning. In the law of copyright this phrase is used in at least two distinct senses. For this reason, the use of the term "substantial similarity," absent some contextual guidance, may be very misleading.

As part of its prima facie case, a plaintiff must prove substantial similarity between copied copyrightable elements of the copyrighted work and expressive elements of the allegedly infringing work. "Substantial similarity," as used in this statement, indicates a degree of similarity between the allegedly infringing material and what is copyrightable (that is, the copyrightable part or parts). It is not enough for a plaintiff to prove great similarity of the allegedly infringing work to uncopyrightable parts of the copyrighted work.

The similarity to copyrightable parts, it is said, must be such that an "ordinary observer" would find that there has been "unlawful appropriation." *Concrete Machinery Co. v. Classic Lawn Ornaments*, 843 F.2d 600, 608 (1st Cir.1988). Thus, the copying must be extensive enough to be "substantial." This is "substantial similarity" in a mixed law-fact evaluative sense. Thus, the test for determining whether there is "substantial similarity" in this sense involves not merely a ministerial task of measuring by a yardstick or word count, but a judgmental task of weighing the quantitative measurements along with other relevant factors and coming to an overall evaluation that applies a legal test for "unlawful appropriation" to the facts.

The plaintiff must also demonstrate copying of the copyrighted work. Unlike patent law, copyright law never determines that an infringement occurs merely because two works are similar, or even identical, so long as the works are independently created. Therefore, to be successful, a plaintiff must demonstrate that the defendant had access to plaintiff's work. Such access is not disputed in this case. Nevertheless, to prove copying, plaintiff must show more than access. Copying can, of course, be proved directly, and there is some direct evidence of copying in this case. However, once access is proved, copying can also be proved by demonstrating "substantial similarity." In this context, "substantial similarity" simply means sufficient similarity of a given element of a work to an element in the allegedly infringing work to support a reasoned inference that more probably than not the element was copied from the copyrighted work. This is not similarity

in a mixed law-fact sense that includes being similar enough to constitute "unlawful appropriation." Rather, the elements of the copyrighted and allegedly infringing work must be shown to be substantially (i.e., notably) similar in a purely factual sense. This is "substantial similarity" in an evidentiary sense. "Substantial similarity" in this sense is one kind of circumstantial evidence of copying.

In this Memorandum, I will try in each instance to indicate clearly (by explicit statement, by context, or by both) to which usage of substantial similarity I am referring.

2. *Consequences of Entanglement*

The complexity introduced by the entanglement of law and fact, bearing on "substantial similarity" (in the mixed law-fact evaluative sense) and on copyrightability has important consequences, including distinctive problems of adjudication and case management both during pretrial proceedings and during trial. Some of these consequences are relevant to the present case.

First, fashioning appropriate verdict forms and instructions to the jury is more difficult when law and fact are so entangled that the usual mode of stating an issue is in the mixed law-fact form.

Second, determining the scope of evidence that is admissible in relation to questions to be decided by the factfinder (jury or trial judge in a nonjury case) is far more difficult. Moreover, failure of the trial judge to resolve problems of admissibility, especially in relation to proffered expert opinion testimony, may substantially impede fair trial because of risks of jury confusion and the incentives to counsel and experts to try to influence the jury with arguments on the law cast in the form of opinions on the mixed law-fact questions.

Third, when the applicable law requires the application of a standard for decisionmaking that is derived from a congressionally mandated accommodation among conflicting public policy interests and thus "reflects a balance of competing claims upon the public interest," *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 431, 104 S.Ct. 774,

783, 78 L.Ed.2d 574 (1984), consequences of the first and second types are enhanced, sometimes beyond measure. Difficult as the task of applying that balance in particular cases may be, however, it is not open to a court to ease the task either by revising that balance, *Feist Publications, Inc. v. Rural Telephone Service Co.*, ___ U.S. ___, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991), or by leaving the jury free to revise that balance. A congressionally mandated balance must be respected by courts and juries alike.

The mixed law-fact evaluative "substantial similarity" issue in copyright law inevitably produces the first and second of these types of consequences, and in particular cases may produce the third as well unless effective measures of judicial control are used. For example, consider whether or not expert opinion evidence about the ways in which two computer software programs are similar and dissimilar should be heard by a jury. One may argue for a negative answer under the authority of *Concrete Machinery Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 608 (1st Cir.1988) (substantial similarity is to be judged under the "ordinary observer" test "unaided by expert testimony"). Even if it may be appropriate and useful to a jury in understanding purely historical facts about characteristics of computer programs to hear expert opinions bearing on historical facts, still opinions that explicitly or implicitly depend on policy premises inconsistent with the law (as determined by the trial judge, subject to correction on appeal) should not be heard by the jury.

Affidavits of experts and arguments of counsel before me in this case illustrate the risks of contentious and potentially unfair proceedings incident to the strong tendency of witnesses and lawyers to intertwine every statement about factual similarity with opinions and arguments about what is, or even should be, the kind of similarity that is legally significant. These opinions are, of course, opinions on the law. Allowing a jury to hear them expressed by witnesses and lawyers is fundamentally inconsistent with the rule that jurors take their instructions on the law solely from the court.

I will return to the issue of "substantial similarity" in the mixed law-fact evaluative sense in Part IV, below.

3. *Copyrightability*

Issues of "copyrightability" in computer software programs present the third type of consequence of entanglement of law and fact in a dimension so extreme that the problem is appropriately viewed as one different in kind and not merely degree. This is one reason, though not the only one, for concluding that at least in the present case, and perhaps more generally, all questions of law and fact bearing on copyrightability of elements of a computer software program are to be decided by a court, not a jury. That issue is considered more fully in Parts IV, VI, and VII, below.

IV. *Entanglement of Copyrightability and Substantial Similarity*

The complexity of entanglement of issues in this case extends not only to issues bearing only on copyrightability, and to issues bearing only on substantial similarity in the mixed law-fact evaluative sense, but also to the interdependence of copyrightability and substantial similarity.

The point is illustrated by Borland's argument that Borland is entitled to summary judgment *on the issue of copyrightability*, Docket No. 141, at 1 (emphasis added), because the elements of Quattro and Quattro Pro that are alleged to be *substantially similar* to elements of Lotus 1-2-3 concern only the way certain parts of the programs function, not copyrightable expressions. *Id.* at 4-8.

Borland argues that *Concrete Machinery*, 843 F.2d at 600, requires that issues of substantial similarity be addressed first, to determine whether "there are sufficient articulable similarities to justify a finding that the defendant has copied from the protected work." Borland's Memorandum, Docket No. 141, at 114 (quoting *Concrete Machinery*, 843 F.2d at 608) (citing *Arnstein v. Porter*, 154 F.2d 464 (2d Cir.1946)). (In this argument, one may note, Borland is using "substantial similarity" in the evidentiary sense.) Then, Borland argues, "the court must determine whether the copying is sufficiently substantial to constitute 'unlawful appropriation' ('illicit

copying')." *Id.* at 114 (quoting *Concrete Machinery*, 843 F.2d at 608). (This argument, it may be noted, focuses both on a strictly factual question about copying and a mixed law-fact question about whether the copying is sufficient to constitute "unlawful appropriation.") "Assuming copying of protected aspects is established, the trier of fact can then assess pursuant to the 'ordinary observer' test whether there is substantial similarity between the protected expression and the accused work." *Id.* (quoting *Concrete Machinery*, 843 F.2d at 609).

The first step formulated in the *Concrete Machinery* opinion may be perceived as itself composed of two parts—determining what are the "protected aspects of the copyrighted work" (that is, the "copyrightability" issue) and determining whether there has been copying of one or more of those aspects. It is not clear that one of these two need precede the other in an analysis of the evidence and arguments presented to a court. Indeed, they may be sufficiently interrelated that in most cases they are best addressed together—a possibility at least suggested by the First Circuit's combining the two into the first "step" of its "two-step" test.

In any event, the argument that First Circuit precedent *requires* a particular order of analysis of a given problem is contrary to a substantial body of precedent in the circuit. For example, the First Circuit spoke quite early to make clear, in the context of discrimination cases, that the factors analysis set forth in *McDonnell-Douglas Corp. v. Greer*, 411 U.S. 792, 93 S.Ct. 1817, 36 L.Ed.2d 668 (1973), is intended as a useful guideline and not a mandatory straitjacket. *See, e.g., Dance v. Ripley*, 776 F.2d 370, 373 (1st Cir.1985) ("This circuit, along with other circuits, has rejected the argument that *McDonnell Douglas* and [*Texas Dept. of Community Affairs v.*] *Burdine* [450 U.S. 248, 101 S.Ct. 1089, 67 L.Ed.2d 207 (1980)] set forth a 'rigid, three-step proof process in Title VII cases.' ") For like reasons, I conclude that the pattern of analysis and decisionmaking presented in *Concrete Machinery* is not meant to be a straitjacket. Thus, I conclude that the order of addressing the various elements of a prima facie case fashioned in *Concrete Machinery* need not be applied to a case in which

there are compelling practical reasons for a different order of proceeding.

Perhaps the procedure discussed in Part V, below, fits the *Concrete Machinery* model because the "first" step of that model is itself two-fold, as noted above. In any event, regardless of whether the procedure discussed in Part V precisely fits the *Concrete Machinery* model, I conclude, for reasons explained in Part V, that it is appropriate to consider an order of proceeding in this case that places issues of substantial similarity (in the evidentiary sense) ahead of issues of copyrightability.

V. *Sharpening the Focus*

In this case, if it is determined that neither a motion for summary judgment nor a motion for directed verdict should be sustained, how might issues involving substantial similarity (in the evidentiary sense) be submitted to the jury in a verdict form in the trial of this case?

To establish infringement, Lotus must prove, either directly or circumstantially, that Borland copied. Circumstantial proof of copying is made by showing access to the copyrighted work and substantial similarity (in the evidentiary sense). In the present case, I conclude that a reasonable trier of fact could find that the Quattro programs' user interfaces were not copied in their entirety from Lotus 1-2-3. However, because access is admitted, the record before me contains both circumstantial proof of copying (e.g., all of the commands in the Lotus 1-2-3 menu structure appear verbatim in the Quattro programs' menu structures in similar hierarchies) and direct proof of copying (e.g., Borland named certain files involved in the 1-2-3 emulation "123.rsc" and "123.mu"). It does not necessarily follow, however, that a jury must on this record find that some particular element was copied from Lotus 1-2-3 rather than being either independently created or copied from some other source.

For any elements of 1-2-3 that Lotus wishes to argue were copied, Lotus must demonstrate (either beyond dispute on motion for summary judgment, or else at trial by a preponderance of the evidence) that the alleged copying from 1-2-3 occurred. Lotus may do so either directly or circumstantially. To demonstrate that the copying was illicit Lotus must also show substantial similarity in the mixed law-fact evaluative sense and copyrightability of those elements.

Returning to the question posed above (How might issues involving substantial similarity in the evidentiary sense be submitted to the jury?) I take note that one option to be considered is to begin with the less complex issue of copying and with a broad focus, then moving toward narrower questions concerning what elements of Quattro or Quattro Pro were copied from 1-2-3.

Would it be appropriate to frame the first question on a Verdict Form in a way such as this (with a similar question to follow for the other allegedly infringing work)?

Question 1

(a) Do you find that the Quattro Pro user interface as a whole was copied from the Lotus 1-2-3 user interface as a whole?[1]

____YES ____NO

(b) Do you find that the part of the Quattro Pro user interface called the "emulation interface" (also called the "1-2-3 compatible interface") as a whole was copied from the Lotus 1-2-3 user interface as a whole?

____YES ____NO

---

[1] It is not entirely clear that Lotus contents that part (a) of Question I should be answered YES, or that Borland contends that a NO answer to part (a) would be material if part (b) were answered YES. Perhaps the parties will agree, or one will persuade the court, that part (a) may be omitted.

(c) If NO, do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 user interface were copied into some part of the Quattro Pro "emulation interface" (also called the "1-2-3 compatible interface")?

(1)	The menu commands	____YES	____NO
(2)	The menu structure	____YES	____NO
(3)	The command sequence	____YES	____NO
(4)	The long prompts	____YES	____NO
(5)	The macro facility	____YES	____NO

To avoid any misunderstanding, I state explicitly that I have not determined that the foregoing draft of Question 1 is appropriate for use in this case. Nor have I determined what, if any, explanatory instructions should be given to the jury along with such a question. Even if a question of this general type is to be used, I will invite submissions by counsel of any different way of framing the question (and associated instructions) and why that different way would be more appropriate. Of course, counsel are invited to submit additional specific elements of alleged similarity about which there is dispute. Also, I do not suggest that all five elements on the present list ought to be included.

In particular, the term "macro facility" is not adequately defined in the parties' present submissions and would require definition. If "macro facility" means some element or characteristic of the user interface distinct from and not including the menu commands, menu structure, and command sequence, Question l(c) might be appropriate as drafted. If, on the other hand, "macro facility" is to be defined in a broader sense that includes the menu command, menu structure, and command sequence elements, as well perhaps as the long prompts, but does not include other aspects of the 1-2-3 interface and for that reason is not the same as "the user interface as a whole,"

the foregoing draft of part (c) might be replaced by the following parts (c) and (d):

(c) Do you find that the Lotus 1-2-3 macro facility as a whole was copied into some part of the Quattro Pro macro facility?

____YES ____NO

(d) Do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 macro facility were copied into some part of the Quattro Pro macro facility?

(1)	The menu commands	____YES	____NO
(2)	The menu structure	____YES	____NO
(3)	The command sequence	____YES	____NO
(4)	The long prompts	____YES	____NO

The purpose of including these alternative drafts in this Memorandum is to begin the process of focusing issues more precisely.

An advantage of beginning a Verdict Form with a question of this type on the subject of copying is that findings of the jury on the various parts of such a question would enable counsel and the court to focus more precisely on (a) the legal elements of "substantial similarity" (in the mixed law-fact evaluative sense) that are material after taking account of the jury findings, and (b) the legal (and factual, if any) elements of copyrightability that are material after taking account of the jury findings. Surely some of the legal and factual issues that cannot be dismissed as immaterial before taking account of the jury findings would be mooted by the findings. For example, if a jury finds that the emulation interface as a whole was copied from the 1-2-3 interface as a whole, copyrightability questions with respect to less of the user interface

than the whole might be moot. On the other hand, if the jury finds that neither the emulation interface nor certain individual elements of it were copied from Lotus 1-2-3, but certain other elements were copied, the material copyrightability questions would be limited, others having been mooted.

I have proposed presenting the question to the jury as one of "copying" rather than "substantial similarity" even though access is undisputed. I have done so because a mere finding of "substantial similarity" in the circumstantial evidentiary sense may be inconclusive on the issue of copying. That is, even if "substantial similarity" in the evidentiary sense creates a presumption of copying, it may be that the presumption is rebuttable and that the jury should be so instructed. If so, Borland might still persuade the jury that the circumstantial inference of copying has been rebutted by extrinsic evidence offered by Borland. I do not at this time decide any of the questions of law suggested in this paragraph.

Consideration may be given to trying Phase One on copying and commencing Phase Two of trial immediately after Phase One, and before the same jury, to address any remaining questions that could properly be submitted to the jury on issues of "substantial similarity" in the mixed law-fact evaluative sense and "copyrightability."

VI. *Framing the Issues of Copyrightability*

A good place to begin the exploration of potential options in this case regarding submission to a jury of issues of copyrightability (if such issues are to be submitted to a jury) is to consider whether it would be appropriate to submit a single copyrightability mixed law-fact question, framed in terms of the applicable legal standard. Before I suggest such a jury question, I first state the standard for deciding copyrightability that will be used in this case, absent further guidance from higher authority before the date of trial.

A. The Standard for Deciding Copyrightability

Having fully considered the submissions now before the court, I conclude that, for reasons stated in the *Paperback* Opinion, 740 F.Supp. at 54-62, and supplemented here, the copyrightability issues in this case should be determined by the legal standard used there and summarized as follows:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, *along the scale from the most generalized conception to the most particularized*, and choose some formulation—some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression.
>
> * * *
>
> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea.
>
> THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."

Id. at 60-61.

In reaching this conclusion, I have fully considered *amicus curiae* briefs filed in this case, with leave of court. Compared with the great majority of *amicus curiae* briefs filed in courts, two of the *amicus* submissions in this case are distinctive: they do not purport to advise the court how the present case should be decided, and they are not filed on behalf of clients who have a special interest aligned with that of one or the other of the parties to the case.[2] The *amicus* submission of the

[2] A third *amicus* brief filed by the Software Entrepreneur's Forum does take the position that Quattro and Quattro Pro ought to be determined not to infringe Lotus' copyright. However, that brief suggests find-

Register of Copyrights concerns issues that I do not reach in this Memorandum, and I do not discuss it further.

The second *amicus* submission was filed by eleven professors of law, teachers of courses in the intellectual property field, who express a concern that the Opinion in *Lotus v. Paperback* and the standard of decision developed in that Opinion, quoted above, give inadequate attention and emphasis to the distinction between a copyrightable expression and a useful process. Although this distinction is noted in the *Lotus v. Paperback* Opinion—*e.g.*, 740 F.Supp. at 53-58—I accept the point that no explicit reference to "process" appears in the standard of decision quoted above. Because I recognize the value of reminding decisionmakers (whether judges or jurors, a point to which I return later in this Memorandum) of the distinction between a useful process and an original expression, I will add a reference to that distinction, restating the standard in the following way:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation—some conception or definition of the "idea," *"system," "process," "procedure," or "method"*—for the purpose of distinguishing between the idea, *system, process, procedure, or method* and its expression.
>
> * * *
>
> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, *system, process, procedure, or method* is limited to elements essential to expression of that idea, *system, process, procedure, or method* (or is one of only a few ways of expressing the idea, *system, process, procedure, or method*) or instead includes identifiable elements of expression not essential to every expression of that idea, *system, process, procedure, or method.*
>
> THIRD, having identified elements of expression not essential to every expression of the idea, *system, process, procedure, or method*, the decisionmaker must focus on whether those *expressive* elements, *taken together*, are a substantial part of the allegedly copyrightable "work."

Id. at 60-61 (added words emphasized, other emphasis deleted). If the answer to the THIRD step is YES, then the expressive elements of the work, taken together, are copyrightable. Copyright protection extends, of course, only to the expressive elements—not to anything more. To demonstrate entitlement to relief, Lotus will be required to prove that Borland copied expressive elements (that is, the particular form of expression and not just the methodology, process, or idea of the user interface of Lotus 1-2-3) and that as a result there is a substantial similarity (in the mixed law-fact evaluative sense) between expressive elements of 1-2-3 and an allegedly infringing Borland program.

I need not and do not tarry over whether this refinement of the stated standard of decision simply makes explicit something that was implicit in *Lotus v. Paperback* or instead is a modification of the standard stated there. In any event, laboring as I am in territory that is uncharted, I conclude that the sources of authority I am bound to respect—constitutional, statutory, and decisional—leave me the choice and perhaps even the responsibility to make this refinement and to do so explicitly.

Apparently, if not explicitly, a premise of Borland's argument is that once some aspect of a "computer program" (as that phrase is used in the statute, 17 U.S.C. § 101) is determined to be a "process" or "system" (as those words are used in the statute, 17 U.S.C. § 102(b)), no part of that functional aspect is copyrightable. (It is not entirely clear whether the argument is that anything "functional" is not "expressive," or instead that even "expressive" elements of anything that is "functional" are not "copyrightable." The choice between

---

rings that ought to be made rather than the legal rule that must be applied. Because the court does not make findings on motions for summary judgment, and even as to other matters addressed in this Memorandum I am not making findings, I do not address that brief further.

these two different ways of phrasing the argument seems, in any event, more linguistic than substantive.) Once functionality is demonstrated, Borland argues, the inquiry about copyrightability ends, and no copyright is possible. This premise of Borland's position is flawed.

The Professors' Amicus Brief might be interpreted as implicitly if not explicitly supporting Borland's premise. In any event, Borland broadly asserts that "all systems are uncopyrightable." Docket No. 141, at 94. Of course, if Borland means only that being a "system" does not make something copyrightable, the point is plainly meritorious but does not help one decide a case such as this one. But Borland appears to be asserting more—that if some part of a computer program is a "system, process, procedure, or method," 17 U.S.C. § 102(b), no copyright of any aspect of that part of the program is possible. If, in so arguing, Borland means that the fact that a computer program is a "system" precludes copyrightability of every part and aspect of that program (or even that if some part of a program is a "system," copyrightability is precluded as to all aspects of that part) the argument is deeply flawed. *See Kieselstein-Cord v. Accessories By Pearl, Inc.*, 632 F.2d 989 (2d Cir.1980) (holding that expressive elements of utilitarian belt buckles were copyrightable). The fallacy becomes apparent as we examine more closely the proposed bright-line rule that "all systems are uncopyrightable." Even source code and object code would not be copyrightable if this rule were the law.

It is worth noting that in this discussion I do not depend upon judicially or academically developed conclusions about fundamental truths, even though the basic nature of copyright law is relevant. Rather, the central point is that because courts are bound by the congressional mandate that something in computer programs is copyrightable, I must reject Borland's premise. Of course, if Congress should at some future time determine that the balance it has struck with respect to the copyrightability of computer programs is not in the best interests of the programming industry and the public, Congress (but not this court) is free to change that balance.

In discussing this issue, as shorthand for convenience only (and with readiness to reconsider should it be suggested that the shorthand has different substantive implications from the longer phrase), I will use the term "process" to refer to "system, process, procedure, or method" as that phrase is used in the statute.

"Process," like "idea," is an abstraction—a creature of the human intellect. A machine may be constructed of pieces of metal, wood plastic, and other materials, put together with nuts, bolts, bearings, and adhesives. A process, on the other hand, is not composed of materials. Instead it is a set of ideas about how to do something. The fact that it is an idea about doing does not make it any less an idea.

Patent law establishes legal rules for process patents that are different in some respects, and not in others, from the rules applying to other patents. The mere fact that patent law allows a means of legal protection for a process, however, does not establish that there is not also some protection in copyright law. Certainly the area of legal protection under the separate legal regimes is not co-extensive, but it is equally clear that there is no particular reason to believe there should never be any area of overlap. Indeed, precedent recognizes some overlap. *E.g., Mazer v. Stein*, 347 U.S. 201, 217, 74 S.Ct. 460, 470, 98 L.Ed. 630 (1954) ("We do hold that . . . patentability . . . does not bar copyright. . . ."). Thus, the fact that a particular computer program may be protected to some extent under one of these legal regimes does not mean it cannot be protected to any extent under the other.

Thinking about copyright protection for computer programs may begin with the point that a person who conceives a "process" shows or tells—uses some form of communication—to make the "process" accessible to another. Showing and telling are modes of expression.

Thus, there is a closely analogous if not precisely identical dilemma about distinguishing an idea from expressions of the idea and a "process" from expressions of the "process." I need not, and do not repeat here the explanation of that dilemma and of statutory and decisional markers that led me in *Lotus*

v. Paperback to the formulation of a standard to be applied to the facts of a particular case to decide whether an aspect of a computer program is a copyrightable expression.

In a broad sense of the term "process" (or the longer phrase for which it stands here), every aspect of a "computer program" is part of a process. Nevertheless, we have a statutory mandate that some aspects of a "computer program" may be copyrightable. Borland and virtually all others who discuss the matter recognize that "source code" and "object code" have copyrightable elements. Yet each "code" is surely a part of a "system, process, procedure, or method." Imagine the response you would receive from a good programmer if you told her that her "source code," or "object code," or both, lacked "system" and "method" and, either taken together or separately, simply could not be regarded as an effective "process" or "procedure" for communicating to the computer what it should do!

Bowing to the congressional mandate and to widely expressed views about the copyrightability of code, Borland—after full explication of a proposed interpretation of *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), and its progeny—continues with the point that the "long-standing copyright principles" applicable to novels, poems, and other traditional literary works have been applied also to computer programs to provide significant protection to software developers.

> For example, under those long-standing principles, *the sequence, structure and organization of the program's code*, in addition to the text of the code itself, *may be protected by copyright*—under the same rationale that protects the detailed plot line and structure of a play in addition to the play's actual dialog.

Docket No. 141 (emphasis added). Borland also concedes, as it must, that this protection of the sequence, structure, and organization may extend to the user interface as well as source code and object code.

Borland's flawed argument (for a bright-line rule that a finding of "process" defeats "copyrightability," regardless of the originality of any particular expression that enables a person other than the creator to use it) must be rejected because it is fundamentally inconsistent with the congressional balance struck in the Copyright Act. The argument is in its nature a one-conclusive-element argument. Expressed in the form of if-then logical operations of the sort a computer might execute upon pairs of binary values, such an argument asks the court to conclude (a) "if process, then not copyrightable," (b) "if idea, then not copyrightable," and (c) "if patent protection for process, then no copyright protection for process." Arguments of this type, urging courts to adopt an overriding rule that one element of a total set of interwoven circumstances be declared the only legally relevant element are like discredited arguments that courts should select one among all the antecedents of an event in human affairs and declare it to be *the* proximate cause. *See* W. Page Keeton et al., *Prosser & Keeton on the Law of Torts* 266 (5th ed. 1984). These are arguments that one aspect of the total circumstances be treated in law as overriding and conclusive. Arguments of this type are often encountered in the legal system, but far less often sustained. Common sense tells us that life and life's experiences are not so neatly separated into discrete categories, without overlap. The legal system aims at reasonable accommodations among conflicting legitimate interests and rarely holds that one such interest totally overrides all others. Moreover, in the present context, respect for the congressional mandates in the Copyright Act requires that Borland's proposed bright-line rule be rejected.

Borland's argument that *Baker v. Selden* adopted a bright-line rule that governs this case is a gross overstatement not only of what the Supreme Court *did* in that case but as well of what the Court *said*. The Court did not there face the issues presented in the present case. Indeed, the Court did not even address issues of expression within the plot, sequence, organization, or structure of artistic works—issues that were

addressed by Learned Hand in his opinions from which I took guidance in deciding *Paperback*.

Attributing to the Supreme Court one's own extrapolations beyond what the Court did and said is a form of reasoning a lower court should not adopt. It is wiser and more in keeping with the judicial role to recognize candidly that we, not the Justices of the Court, are doing the extrapolating. In hearing and deciding *Baker v. Selden* in 1879 the Court did not, and indeed could not, foresee computer programs. The Court was not speaking about computer programs or even about the kinds of issues, more readily inferred to be foreseeable in 1879, that were explicitly addressed in Learned Hand's opinions a half-century later and beyond. Attributing to the Court's decision in 1879 conclusions that counsel or courts may draw today from *Baker v. Selden* by extrapolation is being less than fully candid about the extent to which decisions must be and are being made today. Nor can we attribute to Congress, in its much more recent action, a manifestation of intent drawn from a supposed legislative history arguably appearing in a CONTU report to Congress. Congress speaks primarily by what *Congress says* formally by its enactment. Even though legislative history regarding expressions *by* members of Congress during the legislative process is sometimes used by courts in aid of determining the meaning of the formal enactments of Congress, there is no support for treating as legislative history what another person or entity says *to* Congress. Thus, to be candid about what we are doing and faithful to congressional as well as precedential guidance, we cannot say that *Baker v. Selden*, or the CONTU Commission's interpretation of that decision, established a bright-line rule, "if functional, not copyrightable." As with "causation," we must delve more deeply to understand such concepts as "process," "expression," "copyrightability," and "substantial similarity."

B. Drafting a Jury Interrogatory to Test Its Viability

In deciding copyrightability issues in the present case I will apply the standard formulated in *Lotus v. Paperback*, with the

refinement described above to take account of the concern appropriately expressed in the Professors' Amicus Brief. If "copyrightability" is to be submitted to the jury, then Question 2 of a Verdict Form in which Question 1 concerned copying might read as follows:

Question 2

In answering this question, you are to apply the following legal standard:

FIRST, in making the determination of "copyrightability" of the user interface of 1-2-3 as a whole, or of some element or elements of that interface, focus upon alternatives that counsel may suggest, or that you may conceive, along the scale from the most generalized conception of that user interface, or some element or elements of that interface, to the most particularized, and choose some formulation—some conception or definition of the "idea," "system," "process," "procedure," or "method" of the interface, element, or elements—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

SECOND, focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work." If they are, then you will find those expressive elements, and no more, copyrightable.

Applying this standard, do you find by a preponderance of the evidence that the user interface of Lotus 1-2-3 contained copyrightable elements?

____YES ____NO

As one considers this option, it becomes immediately apparent that its effect is to give the jury virtually uncontrolled discretion. Prohibitions against probing into jury deliberations—*e.g.*, Fed.R.Evid. 606(b)—would preclude courts from determining whether a jury understood and correctly applied the court's explanation of the standard in its charge to the jury. The only judicial control against improper verdicts would be in decisions whether or not to direct a verdict and whether or not to grant a motion for judgment notwithstanding the verdict.

Might counsel and the court devise some greater measure of assurance that the jury understands and faithfully applies congressional directives regarding the scope of copyright protection for computer software programs? Would it be a step in that direction to use the foregoing formulation as part (a) of a question, with something like the following as part (b)?

(b) If YES, is each of the following a part of the user interface in which you find expressive elements, which you also find to be at least part of the basis for your answering YES to Question 2(a)?

(1)	The menu commands	____YES	____NO
(2)	The menu structure	____YES	____NO
(3)	The command sequence	____YES	____NO
(4)	The long prompts	____YES	____NO
(5)	The macro.facility	____YES	____NO
(6)	The overall appearance of some part of the user interface	____YES	____NO

Answers to the subparts of a question of this kind might at least enable a court, after verdict, to give somewhat more reasoned consideration to the relevance and validity of an argument—such as is advanced by Borland here—that, as precedent, *Lotus v. Paperback* should be limited by the finding that Paperback copied the "user interface as a whole," and that the present case is materially different because here, even if the facts are construed most favorably to Lotus, Quattro and Quattro Pro are substantially similar to less of the user interface of Lotus 1-2-3 than was VP Planner. (I do not now determine the validity of this argument.)

Even if an addition such as part (b) might be useful in some circumstances, however, it would not eliminate, or even substantially mitigate, another serious problem to which I turn next.

VII. *Should Any Copyrightability Issues Be Submitted to a Jury?*

Is a fact question bearing upon copyrightability to be submitted to a jury if a timely demand for jury trial has been filed and the fact is genuinely in dispute—that is, reasonable persons might differ about the answer to the question?

Though debated often, as yet this question has not been explicitly answered either by statute or by precedents. Cases that arguably bear upon the question can at the least be distinguished on the facts and thus do not speak directly to the precise kinds of copyrightability issues that are presented in this case. *E.g.*, *Kregos v. Associated Press*, 937 F.2d 700 (2d Cir. 1991) (reversing summary judgment on issue of "creativity"). *Kregos* did not involve the distinctive entanglement problems that are presented in computer-program copyrightability cases.

In *Lotus v. Paperback*, even though the question as to whether copyrightability might be a jury issue had been discussed extensively in pretrial conferences, the court was not required to and did not decide it. The parties removed it from contention by entering into a stipulation. As part of an agree-

ment for a phased trial in which Phase One was to be before the court without a jury, the parties to that case stipulated that any fact question relevant to copyrightability would be decided by the court. No such stipulation has been made in this case, and I now address the issue.

A. Role of Judge, Jury, and Witnesses

The legal test for determining copyrightability both as formulated in *Lotus v. Paperback* and as refined in Part VI, above, is a standard requiring an evaluative mixed law-fact determination, as distinguished from a bright-line rule calling for a finding about disputed historical facts such as who did what, where, and when. Moreover, this standard is far more heavily loaded with public policy implications than most other standards more commonly used in law, of which the negligence standard is an example. Juries applying the copyrightability standard would not be required or even permitted to explain their reasoning. They would be free as a practical matter to reach decisions inconsistent with the balance struck by Congress, as interpreted by the courts. Inconsistencies among verdicts could be expected to introduce a lawless element into the administration of justice in copyright cases, quite inconsistent with the aim of treating like cases alike.

In the circumstances of a particular case, the answer to Question 2 (or any alternatives to Question 2 we might envision), is an essential premise of a reasoned application of the accommodation of conflicting policy interests reached by Congress in determining that computer software programs are copyrightable. For this reason, I conclude that the application of the standard to a particular case is a ruling more closely analogous to traditional judicial lawmaking to fill the interstices of statutes than to traditional factfinding. It is appropriately treated as a ruling of law. Thus, even if a court treats the answer to Question 2 as a "factfinding," the court may conclude also that the answer is a finding of a premise fact—a finding of a fact that serves solely as a premise for a ruling of law. *See* Robert E. Keeton, *Legislative Facts and Similar Things: Deciding Disputed Premise Facts*, 73 Minn. L.Rev. 1

(1988). Like a finding about whether transporting gasoline on a public highway, or blasting with dynamite, or cropdusting, or building an earthen dam is an abnormally dangerous activity, *id.* at 19-20, 57-69, perhaps, this is a finding to be made not by juries, case by case, but by a lawmaker (Congress, to the extent that it can do so in broadly applicable statutory guidelines, and courts to the extent necessary to fill out details essential to application of those broad guidelines in particular types of cases).

The position that this kind of finding is a premise factfinding is supported also by the certainty that even if patterns of jury verdicts might develop over time and become sufficiently predictable to be reasonably described as part of the "law" of copyright, this element of the law would be developed undercover rather than, as is expected of lawmaking through judicial precedents, by "reasoned decisionmaking, candidly explained." Robert E. Keeton, *Judging* 1-2 (1990). Moreover, even if patterns were to develop sufficiently to enable lawyers to predict outcomes for clients with some confidence, still the patterns could not be employed to protect an individual litigant, plaintiff or defendant, against the harsh consequences of a deviant verdict.

The length and complexity of a jury trial of issues of copyrightability would be affected also by disputes regarding admissibility of opinion evidence. The affidavits submitted in this case by the parties, in support of their respective positions on the cross-motions for summary judgment, present what is in essence a clash of policy arguments by experts. Moreover, an impartial factfinder may reasonably infer that the policy positions the experts advance correlate better with their respective views about what the law of copyrightability should be than with considered opinions about the policy accommodation Congress has struck and courts and juries are bound by oath to respect. This point is supported by the following passage from Borland's Memorandum (Docket No. 141):

> More importantly, Borland's witnesses include distinguished industry executives who have years and years

> of actual experience in bringing software products to market. Galler [a Lotus witness], in contrast, can only theorize and speculate about such issues. Unlike Galler, Borland's experts can authoritatively discuss the consequences to software development of the untoward extension of copyright law Lotus seeks in this case.

Id. at 28. Thus, the affidavits of experts submitted to this court for consideration in ruling on the motions for summary judgment are aimed more at persuading this court to a view of the law than to the existence or nonexistence of a genuine dispute of fact.

If copyrightability is held to be an issue for jury determination, courts and counsel must work out answers to extraordinarily difficult issues regarding admissibility of opinion evidence.

No provision of the Copyright Act declares explicitly that issues of fact bearing on copyrightability shall be submitted to juries, or instead shall be decided nonjury. Nor does the Constitution, including the Seventh Amendment. Thus, the answer to the question who shall decide such issues must necessarily be fashioned by courts. Courts are not free, however, to fashion whatever answers they may deem best. They must, instead, seek answers consistent not only with the explicit constitutional and statutory mandates but also with the implications of those mandates for other issues not explicitly addressed. A determination that issues of copyrightability are to be resolved by juries would have such severe adverse effects on the aim of assuring that like cases are treated alike and on the complexity and cost of litigation in computer software copyright cases that in practical effect the scope of copyright protection congress manifestly intended could not be achieved. The practical certainty of many outcomes inconsistent with the congressional accommodation among highly valued but conflicting interests, manifested in the Copyright Act, weighs heavily in favor of the conclusion that treating copyrightability issues as exclusively for courts, not juries, at least in computer software cases, is the decision more compatible with the congressional mandate. So also does the

object of avoiding the practical certainty of increased length, complexity, and cost of litigation.

For all these reasons, I reach the tentative conclusion (which I will be prepared to reconsider as explained in Section B, below) that at least in the circumstances of this case (and probably more generally, though I need not so determine here), the issue or issues of copyrightability, including any fact questions bearing upon them, must be determined by the court, not the jury.

B. Reconsideration

Part II, above, alludes to the value of thinking seriously about the framing of jury questions and instructions well in advance of trial because the attempt to do so helps sharpen the focus on issues that are best considered early in the history of the case. Part VII-A tentatively determines that copyrightability issues in this case are to be determined by the court, not the jury. Before so ruling finally, however, I will allow each party an opportunity to submit a proposed draft of any question bearing on copyrightability that it contends, even in the alternative, is a question that should be submitted to the jury. If the draft question uses legal jargon rather than plain English only, the draft must be accompanied by proposed instructions to the jury explaining the jargon. Submissions and responses may be filed on the schedule set in the Order below.

C. Ambiguity of the Parties' Contention Regarding Copyrightability

It is not now apparent which individual elements of 1-2-3 the plaintiff (Lotus) alleges to be copyrightable, nor has either party made clear its contention about what aspects of various elements of the 1-2-3 user interface are or are not expressive aspects.

Lotus' Statement of Material Facts as to Which There is No Genuine Issue Pursuant to Local Rule 56.1 (Docket No. 149) includes the following paragraphs:

32. The selection of which, and how many, commands to place in each menu level, and the organization of the successive menu levels into a coherent and intuitive menu structure, was an important and creative consideration in the development of 1-2-3. 1-2-3's creators, particularly Mr. Kapor, devoted substantial effort during the late stages of the program's development to selecting the words and structural organization for 1-2-3's menus. (Kapor Aff., ¶¶ 72-101.)

33. Thus, the 1-2-3 menu commands, their organization and sequence, the 1-2-3 menu tree, and the overall user interface reflect an original expression of a spreadsheet program. This expression is not dictated by functional constraints and contains a significant degree of communicative content to the user. (First Galler Dec., ¶ 207.)

Id. at ¶¶ 32, 33.

Borland responds that paragraph 32 is uncontested and immaterial and that paragraph 33 is contested or non-factual. Borland adds, *inter alia*, that "[t]he suggestion in the first sentence of No. 33 that the 1-2-3 commands and command hierarchy represent copyrightable 'expression' is not a statement of fact but rather constitutes an asserted conclusion of law," which Borland disputes. Even though contending that this is an issue of law, Borland has nevertheless resisted trying it separately and first. Thus, I understand Borland's position to be that if the court does not decide this issue for Borland as a matter of law, Borland wishes not to be foreclosed from contending, in the alternative, that copyrightability in general, and this issue in particular, should be submitted to the jury.

I conclude that a decisionmaker (judge or jury) cannot determine the answer to the dispute over Lotus' ¶ 33 without making an evaluative application, to this case, of the standard formulated in Part VI of this Memorandum. One might reasonably argue that an issue such as this, if one for determination by the court, can properly be decided on motion (or

cross-motions) for summary judgment, even if the issue is one on which reasonable persons may differ in the circumstances of this case. Nevertheless, even without so deciding now, I conclude that on the present submissions I am left with a sense that this is a genuinely disputed and debatable issue. Unless one of the parties can persuade me otherwise in further submissions permitted by the attached Order, I will leave this issue to be decided on a full record developed in a first phase of trial, and with the benefit of a sharper focus than present submissions accomplish as to precisely what issues of fact and law should be decisive of this case.

In particular, the dispute between the parties regarding paragraph 33 is one I cannot decide on the present submissions. If the menu commands or menu command structure were dictated solely by functional concerns, then those elements may not be copyrightable. *See Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir.1987) ("conceptual separability" test examines whether expression exercised independent of functional influences). But the mere fact that functional concerns were influential does not establish that no copyrightable expression appears in the menu commands, the command structure, or elsewhere in the user interface.

VIII. *Substantial Similarity Revisited (Illicit Copying)*

It is not enough that an allegedly infringing work contain matter copied from a copyrighted work. Lotus must demonstrate that similarities between the works are substantial. However, not all similarities may be taken into account. For example, the fact that both programs are spreadsheet programs renders them substantially similar in many respects. Nevertheless, such similarities may form no part of a conclusion that the works are substantially similar in the mixed law-fact evaluative sense—that is, for purposes of assessing *illicit* copying—because the idea (or process) of a spreadsheet program is not copyrightable. It is naturally to plaintiff's

advantage to place as many copyrightable elements before the court and jury on the issue of substantial similarity as possible not only because the sum increases with each added element but also because the whole may be more than the sum of its parts. Therefore, a determination as to which elements of Lotus 1-2-3 are copyrightable, alone or together, is interwoven with the issue of substantial similarity.

On the other hand, as I have concluded above, it is by no means apparent that all copyrightable elements of 1-2-3 have been copied in the Quattro programs. Lotus, presenting as its primary contention that *Paperback* is controlling, has not formulated for the court or for Borland its precise contentions (in the alternative to its contention that the user interface as a whole was copyrightable and was copied) as to which elements of 1-2-3, separately or in combination, were copyrightable and were copied.

Borland, on the other hand, at points in its submissions, appears to contend that it is entitled to summary judgment both on the issue of substantial similarity and on the issue of copyrightability because if anything at all in a computer program's user interface is copyrightable it is only the user interface as a whole, and the elements of the Quattro and Quattro Pro user interfaces that are like elements of the Lotus 1-2-3 interface are only a very small part *of the Quattro and Quattro Pro user interfaces*. This is an argument not likely to prevail in any court. Copyrightability is determined by the characteristics of the allegedly protected work, not by the characteristics of the allegedly infringing work. A copier who incorporates a copyrighted short story of twenty pages length into a multi-volume set of books will not get far with the contention, in litigation between the copyright owner of the short story and the copier, that the short story is not copyrightable because it is such a small part of the multi-volume set. Nor will the copier win on the substantial similarity issue by showing that although the entire copyrighted work was copied it constituted only a very small part of the overall set of books produced by the copier.

I need not and do not rule at this time on the appropriate framing of the issues that must be decided in this case about the legal significance that may attach to findings of fact about the disputed scope of similarities between the user interfaces of the different works. To say the least, however, Borland is not entitled to a summary judgment on this extraordinary ground it appears to be asserting.

If Borland responds that I have misinterpreted its submissions and it did not mean to assert such an extreme position, its future submissions regarding precisely what questions it proposes to submit to the jury and what expert opinions it proposes to proffer may clarify its contentions. As to opinion testimony, it seems likely indeed the court should preclude the expression in the presence of a jury of an opinion that "substantial similarity" does not exist when the elements of the user interfaces that are alike constitute only a small percentage *of the allegedly infringing product*. Whether that is correct or not is not a question of fact on which expert opinion may properly be placed before a jury.

The argument that a little copying—beyond "fair use," 17 U.S.C. § 107— here and there in a very large work should be allowed to promote the development of great works is a policy argument Congress has rejected. It should not be heard by a jury. The issue is one of law as to which the court will receive arguments of counsel, outside the presence of the jury, both to determine the scope of admissible opinion testimony and to determine the framing of questions of fact to be submitted to the jury.

Neither, on the other hand, will the jury hear the argument that copying just the menu terms, or just the menu structure, or just a command sequence, is enough to establish infringement. Whether this is so or not is a question of law to be argued to the court, not the jury. To the extent that strictly factual answers from the jury may be helpful to the court and counsel to focus the legal issues to be argued, however, questions may be framed to elicit those answers.

ORDER

For the foregoing reasons, it is ORDERED:

(1) The cross-motions for summary judgment (Docket Nos. 30 and 87) are denied.

(2) Renewed motions for summary judgment may be filed by April 10, 1992 and responses by April 20, 1992.

(3) Submissions of both parties pursuant to Part VI of the foregoing Memorandum shall be filed on the schedule in (2).

(4) The several motions regarding the submission of confidential materials (Docket Nos. 140, 156, and 160) are dismissed as moot.

A conference to consider all pending matters and to set a trial date is scheduled.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 87-76-K

June 28, 1990

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

PAPERBACK SOFTWARE INTERNATIONAL
and Stephenson Software, Limited,

Defendants.

Henry B. Gutman, Kerry L. Konrad, John S. Beckerman, Karen F. Conway, O'Sullivan Graev & Karabell, New York City, and Thomas J. Dougherty, Skadden, Arps, Slate, Meagher & Flom, Boston, Mass., *for Lotus Development Corp.*

Lawrence G. Papale, Cannata, Genovese & Papale, San Francisco, Cal., and Edward C. Saltzberg, Warner & Stackpole, Boston, Mass., *for Stephenson Software, Ltd.*

Paul R. Gupta, David A. Guberman, Barbara O'Donnell, Brian C. Levey, Nereyda F. Garcia, Kenneth R. Berman, Sherin and Lodgen, Boston, Mass., and Walter G. Murphy, Peter C. Kober, Murphy, DeMarco & O'Neil, P.C., Boston, Mass., *for Paperback Software Intern.*

OPINION

KEETON, District Judge.

The expression of an idea is copyrightable. The idea itself is not. When applying these two settled rules of law, how can a decisionmaker distinguish between an idea and its expression?

Answering this riddle is the first step—but only the first—toward disposition of this case in which the court must decide, among other issues, (1) whether and to what extent plaintiff's computer spreadsheet program, Lotus 1-2-3, is copyrightable, (2) whether defendants' VP-Planner was, on undisputable facts, an infringing work containing elements substantially similar to copyrightable elements of 1-2-3, and (3) whether defendants' proffered jurisdictional and equitable defenses are meritorious.

Phase One of this case was tried to the bench. By agreement of the parties, in Phase One, the court shall

> resolve all legal and factual issues concerning the liability, if any, of defendants . . . for the claims of copyright infringement brought by plaintiff . . . and all defenses thereto, including but not limited to all factual and legal issues concerning the copyrightability of Lotus' Works [1-2-3, releases 1.0, 1A, and 2.0], and *excluding only:* (1) issues of fact, if any, requiring jury determination concerning defendants' alleged copying of any protected expression from Lotus' Works in Defendants' Works [VP-Planner and VP-Planner Plus]; [and] (2) factual issues concerning defendants' possible copying of the source or object code for Lotus' Works. . . .

Stipulation and Order Regulating Phased Trial, § I(A) (Docket No. 246).[1]

[1] This case was originally consolidated with *Lotus Development Corporation v. Mosaic Software*, Civil Action No. 87-74-K, which involves Mosaic's computer spreadsheet program The Twin. Although Phase One of this case was intended to resolve certain additional legal and factual issues concerning the liability of Mosaic Software for plaintiff's claims of copyright infringement, that case was severed on the second day of this first phase of trial for reasons not here relevant. *See* Docket No. 174 (February 8, 1990).

This Opinion sets forth findings of fact and conclusions of law that are central to deciding this controversy. Fed.R.Civ.P. 52(a). The court adopts, as additional findings, all proposed findings to which no party objected. *See* Docket Nos. 250, 251.

The outcome of this case depends on how this court, and higher courts on appeal, should answer a central question about the scope of copyrightability of computer programs. For the reasons explained in this Opinion, I conclude that this question must be resolved in favor of the plaintiff, Lotus.

I. A BACKGROUND STATEMENT ABOUT COMPUTERS, COMPUTER PROGRAMS, AND COPYRIGHTABILITY

Though their influence in our society is already pervasive, digital computers—alorg with computer "programs" and "user interfaces"—are relatively new to the market, and newer still to litigation over "works" protected by intellectual property law.

Digital computers (hereinafter referred to as "computers") are machines currently used to perform three types of functions electronically: (1) arithmetic calculations; (2) logical operations (*e.g.*, comparing values to determine whether one is larger); and (3) storage and display of the results. Because computers can perform millions of operations of these types in a single second, they can be used to solve problems too complex, or too repetitious and boring, to be solved manually. Developments to the current state of the art have already transformed many areas of business, educational, and recreational activity, and they support speculations about more striking achievements in the future.

A personal computer system consists of hardware and software. The hardware includes the central processing unit ("CPU"), which contains the electronic circuits that control the computer and perform the arithmetic and logical func-

tions, the internal memory of the computer ("random access memory," or "RAM"), input devices such as a keyboard and mouse, output devices such as a display screen and printer, and storage devices such as hard and floppy disk drives. The software includes one or more computer programs, usually stored magnetically on hard or floppy disks, along with such items as instruction manuals and "templates," which are pieces of plastic that fit around the function keys on the keyboard, identifying the specific functions or commands that can be invoked by those keys. A personal computer system can also include "firmware," or "microcode":

> Microcode is a set of encoded instructions . . . that controls the fine details of the execution of one or more primitive functions of a computer. Microcode serves as a substitute for certain elements of the hardware circuitry that had previously controlled that function.

Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 Duke L.J. 663, 677.

Computer programs are, in general, divided into two types: operating system programs and application programs. Operating system programs—such as DOS, XE-NIX, and OS/2—are programs that control the basic functions of the computer hardware, such as the efficient utilization of memory and the starting and stopping of application programs. Application programs are programs that permit a user to perform some particular task such as word processing, database management, or spreadsheet calculations, or that permit a user to play video games.

This case concerns two competing application programs—Lotus 1-2-3 and VP-Planner—which are primarily spreadsheet programs, but which also support other tasks such as limited database management and graphics creation. Programs such as these, because they can perform several different kinds of tasks, are called "integrated" application programs.

Congress has defined "computer program" as follows:

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

17 U.S.C. § 101 (1988). This "set of statements or instructions," in its literal or written manifestation, may be in the form of object code or source code. It may also be represented, in a partially literal manifestation, by a flowchart. A copyrightable work designed for use on a computer may include, as well, text that appears, for example, in a problem manual or a manual of instructions. These elements of text, however, ordinarily are not referred to in the industry as part of a "computer program" unless they appear on the computer screen and serve a purpose like that of the components of a "help screen" available to a user whenever needed. Elements of this textual type are not at issue in this phase of this case.

Computer programs are typically written in some form of computer programming "language." The "lowest"-level computer programming language is machine language, which is a binary language written in "bits" (*BI*nary digi*TS*). Each bit is equal to one binary decision—that is, to the designation of one of two possible and equally likely values, such as an "on"-"off" or "yes"—"no" choice. These binary decisions, the only kind that a typical computer can understand directly, are commonly represented by 0's and 1's. A sequence of eight bits (which allows 256 unique combinations of bits) is commonly called a "byte" ("by eight"), and 1024 bytes form a "kilobyte" (commonly referred to as "K," *e.g.*, sixty-four kilobytes is "64K"). Machine language may also be represented in hexadecimal form, rather than in binary form, by the characters 0-9 and A-F, where "A" represents 10, "B" represents 11, and so on through "F," which represents "15." In hexadecimal machine language, only two rather than eight characters are required to allow for 256 unique combinations (*e.g.*, 37 instead of 00110111, each of which represents the 55th of 256 combinations; 7B instead of 01111011, each of which represents the 123rd of 256 combinations; EA instead

of 11101010, each of which represents the 234th of 256 combinations). The computer is able to translate these hexadecimal instructions into binary form. Other versions of machine language are represented in decimal (0-9) and octal (0-7) form.

An object program, or object code, is a program written in machine language that can be executed directly by the computer's CPU without need for translation. For example, in the machine language of a certain computer, the instructions to divide the value in "B" by the value in "C" and add that number to the value in "A" may be represented by the following sequence of instructions (in binary form):

0010000000010001; 1000000011010010;
1101000000010000.

An "intermediate"-level programming language is assembly language. Rather than in bits, assembly code is written in simple symbolic names, or alphanumeric symbols, more easily understandable by human programmers. For example, the calculation described above may be represented, in the assembly language of a certain computer, as follows:

LOAD B; DIV C; ADD A.

Because of the primitive nature of assembly language, even relatively simple computations can require long and complex programs.

During the early period of computing, "programmers" ordinarily wrote programs exclusively in machine language. Today, object code is rarely written directly by computer programmers. Rather, modern programmers typically write computer programs in a "higher"-level programming language. These programs are called source programs, or source code. Although "source code" has been defined far more broadly in some of the literature in the field, and in some of the expert testimony in this case, more commonly the term "source code" refers to a computer program written in some programming language—such as FORTRAN (*FOR*mula *TRAN*slation), COBOL (*CO*mmon *B*usiness *O*riented *L*anguage), Pascal, BASIC, or C—that uses complex symbolic

names, along with complex rules of syntax. In a typical higher-level programming language, for example, the above-described computation—that is, (A) + (B/C)—might be represented as follows:

A + B/C.

Unlike machine language, which is unique to each kind of CPU and which is executed directly by the computer, source code programming languages are universal to almost all computers. As a consequence, source code is executed indirectly. Thus, a program written in source code must be translated into the appropriate object code for execution in one type of computer, and into a different object code for execution in another type of computer. The translation can be effectuated by an "interpreter" program or by a "compiler" program. An "interpreter" program is a simultaneous translator that works in conjunction with the application program every time the application program is run, carrying out the instructions of the program one step at a time. In contrast, a "compiler" program translates the program once and for all into machine language, after which the translated program can be executed directly by the CPU without the need for any further resort to the compiler. A distinctive "interpreter" or "compiler" program is available for each type of source code programming language and each type of CPU.

A partly literal and partly pictorial manifestation of a computer program, still farther removed from direct use with the computer, is the flowchart. A flowchart is a graphic representation of a computer program that is written in symbols, rather than in bits or symbolic names, and with a syntax that is graphic rather than grammatical. *See, e.g.*, Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 Harv.L.Rev. 281, 341 n. 235 (1970) (providing simple flowchart). A flowchart can be thought of as a kind of symbolic outline or schematic representation of a computer program's logic, which is written by a programmer once he or she has a conceptualization of the goals of the program. Creating a flowchart (at least, an

early draft) is thus, typically an early phase in the development of a software system, which is followed by the translation of the flowchart into source code. *See* Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan.L.Rev. 1045, 1051 (1989) (identifying five stages in the development of most application programs: (1) defining the desired task; (2) flowcharting; (3) encoding; (4) debugging; and (5) preparing documentation). On computers and computer programs in general, *see* Lewis Affidavit, ¶¶ 18–42 (Docket No. 274); Galler Declaration, ¶¶ 8–20 (Docket No. 257); A. Clapes, *Software, Copyright & Competition: The "Look and Feel" of the Law* 47–64 (1989); R. Bradbeer, P. DeBono & P. Laurie, *The Beginner's Guide to Computers* (1982); *McGraw-Hill Dictionary of Scientific and Technical Terms* (3d ed. 1984).

The parties agree, as a general proposition, that literal manifestations of a computer program—including both source code and object code—if original, are copyrightable. *Stern Electronics, Inc. v. Kaufman*, 669 F.2d 852, 855 n. 3 (2d Cir. 1982) ("written computer programs are copyrightable as literary works"); *Williams Electronics Inc. v. Artic International, Inc.*, 685 F.2d 870, 876–77 (3d Cir.1982) (object code copyrightable); *Hubco Data Products, Corp. v. Management Assistance Inc.*, 219 U.S.P.Q. 450, 454 (D.Id.1983) (same); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir.1983) (source and object code copyrightable), *cert. dismissed*, 464 U.S. 1033, 104 S.Ct. 690, 79 L.Ed.2d 158 (1984); *GCA Corp. v. Chance*, 217 U.S.P.Q. 718, 720 (N.D.Cal.1982) (same); *Midway Manufacturing Co. v. Strohon*, 564 F.Supp. 741, 750 (N.D.Ill.1983) (same); *Digital Communications Associates, Inc. v. Softklone Distributing Corp.*, 659 F.Supp. 449, 454 (N.D.Ga.1987) (same). Also, it appears that flowcharts, if sufficiently detailed and original, are entitled to copyright protection:

> Flowcharts . . . are works of authorship in which copyright subsists, provided they are the product of sufficient intellectual labor to surpass the "insufficient intellectual labor hurdle". . . .

National Commission on New Technological Uses of Copyrighted Works, *Final Report and Recommendations* 43 (1978) hereinafter "*Final Report*"), *reprinted in* 5 *Copyright, Congress and Technology: The Public Record* (N. Henry, ed.1980), *cited with approval in Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1241 (3d Cir.1986), *cert. denied*, 479 U.S. 1031, 107 S.Ct. 877, 93 L.Ed.2d 831 (1987). *See also Synercom Technology, Inc. v. University Computing Co.*, 462 F.Supp. 1003, 1013 n. 5 (N.D.Tex. 1978) (although taking a very narrow view of the scope of copyrightability of nonliteral elements of computer programs, court nevertheless noted in *obiter dictum* that "it would probably be a violation to take a detailed description of a particular problem solution, such as a flowchart . . . and program such a description in computer language") (Higginbotham, J.); *Data Cash Systems, Inc. v. JS & A Group, Inc.*, 480 F.Supp. 1063, 1067 n. 4 (N.D.Ill.1979) (holding that copyright protection extends "to computer programs in their flow chart, source and assembly phases but not in their object phase," court concluded that computer program written in object code was not copyrightable), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir.1980) (program at issue not copyrightable because publication without notice forfeited any copyright protection); *Williams v. Arndt*, 626 F.Supp. 571, 578 (D.Mass.1985) (violation to take detailed prose description and program such description in source code).

Defendants vigorously dispute, however, the copyrightability of any nonliteral elements of computer programs. That is, defendants assert that only literal manifestations of computer programs are copyrightable. Plaintiff, on the other hand, maintains that copyright protection extends to all elements of computer programs that embody original expression, whether literal or nonliteral, including any original expression embodied in a program's "user interface."

One difficulty with plaintiff's argument is the amorphous nature of "nonliteral" elements of computer programs. Unlike the written code of a program or a flowchart that can be printed on paper, nonliteral elements—including such elements as the overall organization of a program, the structure

of a program's command system, and the presentation of information on the screen—may be less tangibly represented. Whether these elements are copyrightable, and if so, how the nonliteral elements that are copyrightable may be identified, are central to deciding this case.

II. CONSTITUTIONAL CONSTRAINTS

In considering the legal issues relevant to whether nonliteral elements of Lotus 1-2-3 are copyrightable, and if so, to what extent, one may appropriately begin with a provision of the Constitution of the United States:

> The Congress shall have Power . . . To promote the Progress of Science . . . by securing for limited Time to Authors . . . the exclusive Right to their . . . Writings. . . .

U.S. Const., Art. I, § 8, cl. 8. The copyright law, codified in Title 17 of the United States Code, rests upon this explicit grant of legislative authority.

Under this constitutional mandate, Congress has broad though not unlimited authority to grant copyright monopolies as needed to promote progress. If Congress were to determine, for example, that copyright protection is unnecessary to "promote the Progress of" computer programming—because, for example, in Congress' view the financial incentives alone of developing new computer programs (without the added benefit of copyright) are enough to encourage innovation, or because incremental innovation might be stifled by expansive copyright protection—then Congress could, without offending the Constitution, provide no copyright protection for computer programs. At the other extreme, were Congress to find that strong copyright protection is necessary to promote the progress of computer programming, Congress could provide for expansive copyright protection for all aspects of computer programs, again without having strayed beyond the bounds of the constitutionally permissible.

Because the constitutional grant of power authorizes Congress to take either path—or to chart some middle course—

this case does not raise constitutional issues. Rather, the issues at stake here are issues of statutory meaning. The central question is not whether Congress could render nonliteral elements such as those of 1-2-3 copyrightable, but whether it has done so. *Banks v. Manchester, Ohio*, 128 U.S. 244, 252, 9 S.Ct. 36, 39, 32 L.Ed. 425 (1888) ("No authority exists for obtaining a copyright beyond the extent to which Congress has authorized it. A copyright cannot be sustained as a right existing at common law; but, as it exists in the United States, it depends wholly on the legislation of Congress.").

III. CONGRESSIONAL MANDATES AND JUDICIAL INTERPRETATION

A. Sources of Guidance

1. Precedent on Determining Statutory Meaning

In *Kelly v. Robinson*, 479 U.S. 36, 107 S.Ct. 353, 93 L.Ed.2d 216 (1986) (Powell, J., joined by Rehnquist, C.J., and Brennan, White, Blackmun, O'Connor, and Scalia, JJ.), the Supreme Court explained the various sources of guidance to which a court should look in determining the manifested meaning of a statute and the manifestations of congressional intent:

> [T]he "starting point in every case involving construction of a statute is the language itself." *Blue Chip Stamps v. Manor Drug Stores*, 421 U.S. 723, 756 [95 S.Ct. 1917, 1935, 44 L.Ed.2d 539] (1975) (Powell, J., concurring). But the text is only the starting point. As Justice O'Connor explained last Term, " ' "In expounding a statute, we must not be guided by a single sentence or member of a sentence, but look to the provisions of the whole law, and to its object and policy." ' " *Offshore Logistics, Inc. v. Tallentire*, 477 U.S. 207, 221 [106 S.Ct. 2485, 2493, 91 L.Ed.2d 174] (1986) (quoting *Mastro Plastics Corp. v. NLRB*, 350 U.S. 270, 285 [76 S.Ct. 349, 359, 100 L.Ed. 309] (1956) (in turn quoting *United States v. Heirs of Boisdoré*, 8 How. [49 U.S.] 113, 122 [12 L.Ed. 1009] (1850))).

Id. at 43, 107 S.Ct. at 358. Accordingly, to determine whether Congress has extended copyright protection to nonliteral elements of computer programs, and if so, to what extent, a court must examine, first, the relevant language of the copyright statutes, second, "the provisions of the whole law," and third, "its object and policy." *Id.*

Examination of these sources exposes mandates inconsistent with the principal argument advanced by defendants—that copyright protection extends only to literal manifestations of computer programs and not to any nonliteral elements. Close examination also discloses that Congress has not explicitly addressed some of the questions that must be decided in this case. In these circumstances, it is appropriate to consider legislative history as well as statutory text.

2. The Statutory Language and the History of Amendments

a. Pre-1976 Legislation

The First Congress extended copyright protection to "any map, chart, book or books already printed." Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124, 124 (repealed 1831). Congress quickly expanded this limited scope of copyright protection, adding designs, prints, etchings and engravings in 1802, "musical composition" in 1831, "dramatic composition" in 1856, "photographs and the negatives thereof" in 1865, and "statuary" and "models or designs intended to be perfected as works of the fine arts" in 1870. Act of Apr. 29, 1802, ch. 36, § 2, 2 Stat. 171, 171, *repealed by* Act of Feb. 3, 1831, ch. 16, §§ 1, 14, 4 Stat. 436, 436, 439, *amended by* Act of Aug. 18, 1856, ch. 169, 11 Stat. 138, 139, *amended by* Act of Mar. 3, 1865, ch. 126, §§ 1, 2, 13 Stat. 540, 540, *repealed by* Act of July 8, 1870, ch. 180, § 86, 16 Stat. 198, 212 (repealed 1909).

In 1909, Congress abandoned the effort to list exhaustively all works in which copyright may subsist, instead adopting a more generalized approach to copyrightability:

> The works for which copyright may be secured under this title shall include *all the writings of an author.*

Act of Mar. 4, 1909, ch. 320, § 4, 35 Stat. 1075, 1076 (emphasis added) (previously codified at 17 U.S.C. § 4, *reprinted in* 17 U.S.C.A. App. § 4 (West Supp.1990); recodified 1947; repealed 1976). To clarify the meaning of "all the writings of an author," Congress also provided a non-exclusive list of examples. *Id.* at § 5 (listing the various kinds of works previously entitled to explicit copyright protection). Even this generalized statement of the scope of copyrightability along with the explanatory list proved inadequate. In 1912, Congress added "motion pictures" as a further example of "all the writings of an author." Act of Aug. 24, 1912, ch. 356, § 5(1)–(m), 37 Stat. 488, 488 (previously codified at 17 U.S.C. § 5(1)–(m), *reprinted in* 17 U.S.C.A. App. § 5(1)–(m); recodified 1947; repealed 1976), and in 1972, Congress added "sound recordings" to the list, Act of Oct. 15, 1971. Pub.L. 92-140, § 1(b), 85 Stat. 391, 391 (previously codified at 17 U.S.C. § 5(n), r*eprinted in* 17 U.S.C.A. App. § 5(n); repealed 1976).

In 1955, Congress began to consider another major revision of the copyright law. After twenty years of hearings, study, debate, and redrafting, that revision was signed into law in 1976. Act of Oct. 19, 1976, Pub.L. 94-553, 90 Stat. 2541 (codified at 17 U.S.C. §§ 101 *et seq.).* That law—the Copyright Act of 1976—and the 1980 amendments to the Act provide the relevant statutory mandates for this case.

b. The Copyright Act of 1976

Like the Copyright Act of 1909, the Copyright Act of 1976 eschews the prescription of an exclusive list of the kinds of works that are copyrightable:

> Copyright protection subsists, in accordance with this title, in *original works of authorship* fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.

17 U.S.C. § 102(a) (1988) (emphasis added).

It is axiomatic that the designation "original" is not intended to be limited to works that are novel or unique. Rather, the word "original," which was "purposely left undefined" by Congress, refers to works that have been "independently created by an author," regardless of their literary or aesthetic merit, or ingenuity, or qualitative value. H.R.Rep. No. 1476, 94th Cong., 2d Sess. 51, *reprinted at* 1976 U.S. Code Cong. & Admin. News 5659, 5664 (hereinafter "House Report"); *Hutchinson Telephone Co. v. Fronteer Directory Co.*, 770 F.2d 128, 131 (8th Cir.1985); *Puddu v. Buonamici Statuary, Inc.*, 450 F.2d 401, 402 (2d Cir.1971) ("originality has been considered to mean 'only that the work owes its *origin* to the author' ") (emphasis added; *quoting* Nimmer, *Copyright* § 10 at 32 (1971 ed.)). *See also Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 251, 23 S.Ct. 298, 300, 47 L.Ed. 460 (1903) (Holmes, J.) ("It would be a dangerous undertaking for persons trained only in the law to constitute themselves final judges of the worth of" a work.).

Also, the designation "works of authorship" is not meant to be limited to traditional works of authorship such as novels or plays. Rather, Congress used this phrase to extend copyright to new methods of expression as they evolve:

> The history of copyright law has been one of gradual expansion in the types of works accorded protection, and the subject matter affected has fallen into two general categories. In the first, scientific discoveries and technological developments have made possible new forms of creative expression that never existed before. In some of these cases the new expressive forms—electronic music, filmstrips, and *computer programs*, for example—could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and were thus considered copyrightable from the outset without need of new Legislation. In other cases, such as photographs, sound recordings, and motion pictures, statutory enactment was deemed necessary to give them full recognition as copyrightable works.

> Authors are continually finding new ways of expressing themselves, but it is impossible to foresee the forms that these new expressive methods will take. The bill does not intend either to freeze the scope of copyrightable subject matter at the present stage of communications technology or to allow unlimited expansion into areas outside the present congressional intent.

House Report at 51, *reprinted at* 5664 (emphasis added).

To help illumine the meaning of "works of authorship," Congress, as it had done in the 1909 Act, again provided a statutory list of examples of those kinds of works that, if original, merit copyright protection:

> Works of authorship include the following categories: (1) literary works; (2) musical works, including any accompanying words; (3) dramatic works, including any accompanying music; (4) pantomimes and choreographic works; (5) pictorial, graphic, and sculptural works; (6) motion pictures and other audiovisual works; and (7) sound recordings.

17 U.S.C. § 102(a) (1988). This listing was intended to be " 'illustrative and not limitative,' and . . . the seven categories do not necessarily exhaust the scope of 'original works of authorship' that the bill is intended to protect." House Report at 53, *reprinted at* 5666. Consequently, in addition to these explicitly-listed items, courts have extended copyright protection to such works as artistic features of masquerade costumes, *National Theme Productions, Inc. v. Jerry B. Beck, Inc.*, 696 F.Supp. 1348, 1354 (S.D.Cal.1988); the arrangement of public-domain legal decisions in reporters, *West Publishing Co. v. Mead Data Central, Inc.*, 799 F.2d 1219 (8th Cir.1986), *cert. denied*, 479 U.S. 1070, 107 S.Ct. 962, 93 L.Ed.2d 1010 (1987); telephone books, *Southern Bell Telephone & Telegraph Co. v. Associated Telephone Directory Publishers*, 756 F.2d 801 (11th Cir.1985); televised news reports. *Pacific and Southern Co. v. Duncan*, 744 F.2d 1490, 1494, *reh'g denied*, 749 F.2d 733 (11th Cir.1984), *cert. denied*, 471 U.S. 1004, 105 S.Ct. 1867, 85 L.Ed.2d 161

(1985); blank answer sheets for use with student achievement and intelligence tests that are designed to be corrected by optical scanning machines, *Harcourt, Brace & World, Inc. v. Graphic Controls Corp.*, 329 F.Supp. 517 (S.D.N.Y.1971) (applying the 1909 Act); maps, *C.S. Hammond & Co. v. International College Globe, Inc.*, 210 F.Supp. 206, 216 (S.D.N.Y. 1962) (applying the 1909 Act); and a code book for cable correspondence consisting of 6,325 coined, otherwise meaningless, words of five letters each, *Reiss v. National Quotation Bureau, Inc.*, 276 F. 717 (S.D.N.Y.1921) (L. Hand, J.) (applying the 1909 Act).

Although Congress did not include "computer programs" in this list of examples of "works of authorship," computer programs fall squarely within the statutory definition of literary works:

> "Literary works" are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

17 U.S.C. § 101 (1988). *See also* House Report at 54, *reprinted at* 5667 ("[t]he term 'literary works' . . . includes . . . computer programs") *id.* at 51, *reprinted at* 5664 ("computer programs . . . were . . . considered copyrightable from the outset"); *id.* at 116, *reprinted at* 5731 (1976 Act governs "copyright-ability [sic] of computer programs").

Like all other works of authorship, however, computer programs, even if certain elements of them are copyrightable, are not entitled to an unlimited scope of copyright protection. Most relevant to this case is the following limitation:

> In no case does copyright protection for an original work of authorship extend to an *idea, procedure, process, system, method of operation, concept, principle, or discovery*, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b) (1988) (emphasis added). Noting that this section applies to computer programs, the House Report declares: "Section 102(b) is intended, among other things, to make clear that the *expression* adopted by the programmer is the copyrightable element in a computer program, and that the actual *processes or methods* embodied in the program are not within the scope of the copyright law." *House Report* at 57, *reprinted at* 5670 (emphasis added); *see also id.* at 54, *reprinted at* 5667 (computer programs are copyrightable only "to the extent that they incorporate authorship in the programmer's *expression* of original ideas, as distinguished from the *ideas* themselves") (emphasis added).

c. CONTU and the 1980 Amendments

Most contemporaneous observers expected that the bill that eventually became the Copyright Act of 1976 would apply to computer programs. Toward the end of the twenty-year process of reconsidering the copyright law, however, Congress recognized that certain problems raised by computer and other new technologies were not adequately addressed in the pending bill. *See* House Report at 116, *reprinted at* 5731. Accordingly, in 1974, Congress created the National Commission on New Technological Uses of Copyrighted Works ("CONTU"). Congress gave the Commission the following mandate:

> (b) The purpose of the Commission is to study and compile data on:
>
> (1) the reproduction and use of copyrighted works of authorship—
>
> (A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information . . . .
>
> (c) The Commission shall make recommendations as to such changes in copyright law or procedures that may be necessary to assure for such purposes access to copyrighted works, and to provide recognition of the rights of copyright owners.

Act of Dec. 31, 1974, Pub.L. 93-573, § 201(b)-(c), 88 Stat. 1873, 1873-74 (1974).

It took seven months after enactment of the bill to constitute the Commission and appoint the Commissioners. Because of this delay, CONTU did not begin its deliberations until October 1975, and did not release its report and recommendations until July 1978, almost two years after the passage of the 1976 Act.

CONTU observed a need for copyright protection of creative expression embodied in computer programs:

> The cost of developing computer programs is far greater than the cost of their duplication. Consequently, computer programs . . . are likely to be disseminated only if . . . [t]he creator can spread its costs over multiple copies of the work with some form of protection against unauthorized duplication of the work. . . . The Commission is, therefore satisfied that some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market, . . . [and] that the continued availability of copyright protection for computer programs is desirable.

CONTU, *Final Report* at 20-21. Concluding, however, that the Act of 1976 already provided adequate protection, CONTU did not propose any statutory changes with respect to *copyrightability* of computer programs. On the other hand, CONTU did propose two amendments with respect to *permissible copying* of computer programs. These proposed amendments were subsequently adopted by Congress with only minor modifications and with little additional legislative history.

First, tracking verbatim CONTU's recommendation, Congress amended section 101 to include the following definition:

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

Act of Dec. 12, 1980, Pub.L. 96-517, 94 Stat. 3015, 3028 (codified at 17 U.S.C. § 101). Second, Congress also followed CONTU's recommendation by amending section 117 to allow the owner of a computer program to make additional copies or adaptations of the program:

> Notwithstanding the provisions of section 106 [which grants the copyright owner the exclusive rights to reproduce the copyrighted work], it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
>
> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
>
> (2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.
>
> Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

Id. (codified at 17 U.S.C. § 117).

Finally, although the Commission did not address explicitly the central issue of the present case—whether copyright protection extends to the nonliteral elements of computer programs at issue here—CONTU did re-emphasize, for purposes of copyrightability of computer programs, the fundamental distinction between copyrightable expression on the one hand, and noncopyrightable methods, processes and ideas on the other. *Id.* at 37-46. Indeed, although his personal views are entitled to very little if any weight in the context of the

court's determination of the statutory mandates, *Whelan*, 797 F.2d at 1241, n. 37, it is interesting to note that Melville Nimmer, Vice-Chairperson of CONTU, testified about the Commission's intent. in *Evergreen Consulting v. NCR Comten, Inc.*, No. 82-5946-KN (C.D.Cal. filed 1982). According to Nimmer, CONTU understood that the proposed repeal of former section 117 would extend copyright protection to nonliteral elements of computer programs:

> CONTU had no views, and made no recommendations which would negate the availability of copyright protection for the detailed design, structure and flow of a [computer] program under the copyright principles that make copyright protection available, in appropriate circumstances, for the structure and flow of a novel, a play or a motion picture.

Nimmer Decl. in *Evergreen Consulting*, ¶ 28, *reprinted in* Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich.L.Rev. 866, 889 (1990). *But cf.* conflicting views of Commissioner Arthur Miller and Executive Director Arthur Levine, discussed in *id.* at 888-90.

3. Relevant Aspects of the Whole Law of Copyright

a. *"Nonliteral" Expression*

With respect to such things as musical, dramatic, and motion picture works, and works of "literature" (as contrasted with "literary" works in the broader statutory sense, *see* Samuelson, *Reflections on the State of American Software Copyright Law and the Perils of Teaching It*, 13 Colum.-VLA J.L. & Arts 61, 65 n. 15 (1988)), it is crystal clear that, to the extent original, the literal manifestations of such works are protected by copyright. Thus, during a period of copyright protection, one cannot copy an author's book, score, or script without authorization in law or in fact. It is also well settled that a copyright in a musical, dramatic, or motion picture work, or a work of literature, may be infringed even if the infringer has not copied the literal aspects of the work. That

is, even if an infringer does not copy the words or dialogue of a book or play, or the score of a musical work, infringement may be found if there is copying of the work's expression of setting, characters, or plot with a resulting substantial similarity. *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir.1930) (L. Hand. J.) ("the right cannot be limited literally to the text, else a plagiarism would escape by immaterial variations"). *cert. denied*, 282 U.S. 902, 51 S.Ct. 216, 75 L.Ed. 795 (1931); *Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49, 55 (2d Cir.) (L. Hand. J.) ("a play may be pirated without using the dialogue"), *cert. denied*, 298 U.S. 669, 56 S.Ct. 835, 80 L.Ed. 1392 (1936); *Detective Comics, Inc., v. Bruns Publications, Inc.*, 111 F.2d 432, 433 (2d Cir.1940) (defendant's comic book "Wonderman" adjudged to infringe plaintiff's copyright in the comic book series "Superman" where both comic books' central characters have miraculous strength and speed; conceal their strength, along with their skin-tight acrobatic costumes, beneath ordinary clothing; are termed champions of the oppressed; crush guns; stop bullets; and leap over or from buildings); *Bradbury v. Columbia Broadcasting System, Inc.*, 287 F.2d 478, 482-84 (9th Cir.) (twenty-two nonliteral similarities between plaintiff's *Fahrenheit 451* and *The Fireman* and defendants' television production), *cert. dismissed*, 368 U.S. 801, 82 S.Ct. 19, 7 L.Ed.2d 15 (1961); *Roth Greeting Cards v. United Card Co.*, 429 F.2d 1106 (9th Cir.1970) (substantial similarity between nonliteral expressive elements embodied in defendant's cards and plaintiff's cards); *Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp.*, 562 F.2d 1157 (9th Cir.1977) (defendant's television commercial substantially similar in locale, characters, and plot to plaintiff's children's television series); *Twentieth Century-Fox Film Corp. v. MCA, Inc.*, 715 F.2d 1327 (9th Cir.1983) (thirteen alleged similarities between plaintiff's motion picture, *Star Wars*, and defendant's motion picture and derivative television series, *Battlestar: Gallactica*—including totality of setting, characters and their relationships, and elements of plot—were sufficient to create genuine dispute of material fact to defeat

defendant's motion for summary judgment); *Horgan v. Macmillan, Inc.*, 789 F.2d 157, 162 (2d Cir.1986) (copyright in choreography for *The Nutcracker* ballet may be infringed by a book of photographs of the ballet if the series of photographs is substantially similar to the ballet). *See also Stewart v. Abend*, ___ U.S. ___, 110 S.Ct. 1750, 1759, 109 L.Ed.2d 184 (1990) (noting that a motion picture may infringe a book by using "the story's unique setting, characters, plot, and sequence of events"). This type of copying of nonliteral expression, if sufficiently extensive, has never been upheld as permissible copying; rather, it has always been viewed as copying of elements of an expression of creative originality.

b. *"Useful Articles"*

A "useful article" is defined by the Copyright Act as

> an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a "useful article."

17 U.S.C. § 101 (1988). Such articles—or more accurately, the utilitarian aspects of such articles—are not works of authorship in which copyright can subsist. House Report at 55, *reprinted at* 5668. However:

> [T]he design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work [which *is* copyrightable] . . . if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.

17 U.S.C. § 101 (1988) (defining "pictorial, graphic, and sculptural works"). Put more broadly, the point is that those elements of a useful article that can exist independently of the utilitarian aspects of the article are potentially copyrightable because those elements are elements of *expression* that can be distinguished from the utilitarian functions of the article. *See*

Whelan, 797 F.2d at 1236 ("the purpose or function of a utilitarian work [is] the work's idea, and everything that is not necessary to that purpose or function [is] part of the expression of the idea"); *Williams Electronics, Inc. v. Bally Manufacturing Corp.*, 568 F.Supp. 1274, 1279-80 (N.D Ill.1983) (Congress intended to exclude from copyright protection functional elements of work, leaving only those aspects of work separable from utilitarian aspects copyrightable).

4. The Objects and Policies of Copyright Law

The court's final task in divining the statutory mandates is to look to the "object and policy" of the copyright law. *Kelly*, 479 U.S. at 43, 107 S.Ct. at 358. This inquiry has received heightened attention in this case *(see* Part VII, *infra*) because defendants contend that extending copyright protection to nonliteral elements of computer programs is contrary to the objects and policies of copyright law as expressed in the copyright statute and in precedents. Although, of course, disputing defendants' conclusion, plaintiff does not contest the premise of this contention—that in construing the manifested meaning of the Copyright Act, the court is directed to look to the "object and policy" of the copyright law.

Copyright monopolies are not granted for the purpose of rewarding authors. Rather, Congress has granted copyright monopolies to serve the public welfare by encouraging authors (broadly defined) to generate new ideas and disclose them to the public, being free to do so in any uniquely expressed way they may choose. *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 546, 105 S.Ct. 2218, 2223, 85 L.Ed.2d 588 (1985). As the Supreme Court has concluded, "encouragement of individual effort by personal gain is the best way to advance public welfare through talents of authors and inventors in 'Science and useful Arts.' " *Mazer v. Stein*, 347 U.S. 201, 219, 74 S.Ct. 460, 471, 98 L.Ed. 630 *reh'g denied*, 347 U.S. 949, 74 S.Ct. 637, 98 L.Ed. 1096 (1954). *See also Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 432, 104 S.Ct. 774, 783, 78 L.Ed.2d 574 (1984) (" 'The immediate effect of our copyright

law is to secure a fair return for an "author's" creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good.' ") *(quoting Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156, 95 S.Ct. 2040, 2043, 45 L.Ed.2d 84 (1975)).

In construing the relevant statutory mandates, the court must be faithful to the statutory language and mindful of both the ultimate goal of copyright law—the advancement of public welfare—and Congress' chosen method of achieving this goal—private reward to the individual author. Courts should not draw the line between copyrightable and non-copyrightable elements of computer programs in such a way as to harm the public welfare, nor should courts ignore the accommodation struck by Congress in choosing to advance the public welfare by rewarding authors. These mandates leave courts with a delicate task:

> Drawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products. Drawing the line too conservatively would allow programmer's efforts to be copied easily, thus discouraging the creation of all but modest incremental advances.

Menell, *Scope of Copyright Protection for Programs*, 41 Stan.L.Rev. at 1047-48. *See also* Note, *Scope of Copyright Protection of Computer Program Structure*, 88 Mich.L.Rev. at 895; Note, *Defining the Scope of Copyright Protection for Computer Software*, 38 Stan.L.Rev. 497, 498 (1986).

Rather than itself drawing the boundary line between copyrightable and non-copyrightable elements of computer programs, Congress has mandated that courts use an evaluative standard in determining this boundary line—that is, a standard that distinguishes idea from expression and requires that a court, in applying this distinction, be sensitive to the object and policy of copyright law as manifested by Congress.

When statutes establish evaluative standards for deciding cases, courts—by necessity—must locate boundaries in unchartered terrain, using the markers that Congress has placed. At some places a boundary may run straight as a surveyor's sight-line between markers; at others, it may meander like a stream, moving toward resolution of clashing objects and policies the markers identify. No marker Congress has placed may be disregarded or relocated by courts. Even in those instances where text and context make clear that literal description of a marker is contrary to manifested meaning (as where "not" must be inserted or deleted to make sense of the statutory language) what the court is doing is aptly described as "locating," not "relocating," the marker Congress mandated. The fewer the markers Congress has placed, the more critical it becomes that courts assure that no marker escapes notice.

B. The Idea-Expression and Useful-Expressive Distinctions

Although the statutory mandates are ambiguous in some critical respects, one point on which they are clear (one marker of the boundary line) is this: at least some, but clearly not all, aspects of computer programs, if original, are "works of authorship" in which copyright can subsist. 17 U.S.C. § 102(a)(1); House Report at 54, reprinted at 5667; CONTU, *Final Report* at 21. How can a court determine which aspects are copyrightable?

The interplay between sections 102(a) and 102(b), illumined by the related legislative history, manifests that the statute extends copyright protection to expressive elements of computer programs, but not to the ideas, processes, and methods embodied in computer programs. House Report at 54, 57, *reprinted at* 5667, 5670. This dichotomy—which is often referred to as the "idea-expression distinction," and which embraces also the process-expression, method-expression, and useful-expressive distinctions, *see* Note, *Determining the Scope of Copyright Protection of Computer Program Structure*, 88 Mich.L.Rev. at 866-67—has long been a fundamental part of our copyright law. *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879). In that seminal case, the Court held that the

text of a book describing a special method of double-entry accounting on paper spreadsheets—the now almost universal T-accounts system—was copyrightable *expression*, but that the method itself, which embodied the *idea* of this particular kind of double-entry bookkeeping, was not. The Court thus concluded that Baker did not infringe Selden's copyright when Baker wrote his own treatise, in his own words, describing the special double-entry method of bookkeeping.

CONTU, too, concluded that the idea-expression distinction should be used to determine which aspects of computer programs are copyrightable. CONTU, *Final Report* at 37-46. The Commission recognized, though, that "[i]t is difficult, either as a matter of legal interpretation or technological determination, to draw the line between the copyrightable element of style and expression in a computer program and the process which underlies it." *Id.* at 44.

CONTU, of course, was not an official voice of Congress, and its views are not, without more, attributable to Congress. Thus, courts must not treat the CONTU report as legislative history, in the ordinary sense, much less as an authoritative statement about manifested legislative intent. *Whelan*, 797 F.2d at 1241-42. The privately held views of Vice-Chairperson Nimmer and Commissioner Miller, and especially the privately held views of Executive Director Levine, *see* declarations in *Evergreen Consulting*, without some manifestation of congressional endorsement, are even less relevant to the determination of manifested congressional intent. *Id* at 1241 n. 37.

Congress, however, did not ignore CONTU. Indeed, as already noted, Congress adopted practically verbatim the Commission's proposed statutory changes with respect to computer programs. Thus, the expressed views of the Commission, to the extent not repudiated by Congress, may help to explain the context in which Congress acted, which in turn may support inferences about the meaning of any otherwise ambiguous passages in what Congress declared.

I conclude that, with the Copyright Act of 1976 and the 1980 amendments to that Act, Congress manifested an intention to use the idea-expression distinction as part of the test of copyrightability for computer programs. That is, rather than adopting some other test of copyrightability that made the idea-expression dichotomy irrelevant (such as defendants' proposed literal-nonliteral distinction), Congress chose to extend copyright protection to original expression embodied in computer programs, but not to any idea, method, or process described by that expression. 17 U.S.C. §§ 102(a), 102(b); House Report at 54, 57, *reprinted at* 5667, 5670.

That conclusion is consistent with the treatment of other kinds of intellectual works—specifically, with the treatment of nonliteral elements of expression in musical, dramatic, and motion picture works, and works of literature. It is also consistent with the treatment of useful articles. That is, as explained more fully in Part IV(A), *infra*, I conclude that the user interface and some other nonliteral aspects of computer programs are *not merely* articles "having an intrinsic utilitarian function." 17 U.S.C. § 101 (defining "useful article"). When computer programs include elements—both literal and nonliteral—"that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article," *id.* (defining "pictorial, graphic, and sculptural works"), they are potentially copyrightable.

Finally, this conclusion is also consistent with the objects and policies of copyright—to encourage the creation and dissemination of new *ideas* by protecting, for limited times, the specific way that an author has *expressed* those ideas.

IV. THE LEGAL TEST FOR COPYRIGHTABILITY APPLICABLE TO THIS CASE

A. *Functionality, Useful Articles, and the Useful-Expressive Distinction*

Defendants suggest that the user interface of Lotus 1-2-3 is a useful, "function[al]" object like the functional layout of gears in an "H" pattern on a standard transmission, the functional assignment of letters to keys on a standard QWERTY keyboard, and the functional configuration of controls on a

musical instrument (*e.g.*, keys of a piano). Lewis Affdvt. ¶¶ 52-54. These "functional" "useful articles," defendants contend, are not entitled to copyright protection.

A similar analogy was made in *Synercom* where the court concluded that a sequence of data inputs for a statistical analysis program was like the "figure-H" pattern of a standard transmission. 462 F.Supp. at 1013. *Synercom,* though, was published less than a month after the publication of the CONTU report (which it never cites) and well before the 1980 amendments. Since then, congressional and judicial development of the law of copyrightability of computer programs has advanced considerably, and *Synercom's* central proposition—that the expression of nonliteral sequence and order is inseparable from the idea and accordingly is not copyrightable—has been explicitly rejected by several courts. *E.g., Whelan*, 797 F.2d at 1240, 1248 ("copyright protection of computer programs may extend beyond the programs' literal code to their structure, sequence, and organization"); *Broderbund Software, Inc. v. Unison World, Inc.*, 648 F.Supp. 1127, 1133 (N.D.Cal.1986) ("copyright protection is not limited to the literal aspects of a computer program, but rather . . . it extends to the overall structure of a program, including its audiovisual displays"). Moreover, even those courts that have not gone as far as *Whelan* and *Broderbund* have still gone much farther in protecting computer programs than *Synercom. E.g., SAS Institute, Inc. v. S & H Computer Systems, Inc.*, 605 F.Supp. 816, 830 (M.D.Tenn.1985) ("copying of the organization and structural details" can form basis for infringement); *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F.Supp. 984, 993 (D.Conn.1989) ("screen displays or user interface" copyrightable); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir.1989) (nonliteral aspects such as "structure, sequence and/or organization of the program, the user interface, and the function, or purpose, of the program", are copyrightable to the extent that they embody expression rather than idea); *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q.2d 1991, 1993, 1989 WL 200350 (N.D. Cal.1989) (holding that "[c]opyright protection applies to the user interface, or overall structure and organization of a computer program, including its audiovisual displays, or screen 'look and feel,' " but finding no infringement in this case); *Q-Co. Industries v. Hoffman*, 625 F.Supp. 608, 615-16 (S.D.N.Y.1985) (similarity of "structure and arrangement" can form basis of infringement suit, but here, structural similarities were dictated by functional considerations and hence were non-copyrightable ideas rather than copyrightable expression); *Pearl Systems, Inc. v. Competition Electronics, Inc.*, 8 U.S.P.Q.2d 1520, 1524, 1988 WL 146047 (S.D.Fla.1988) ("Copyright protection of computer software is not limited to the text of the source or object code"). *But see Softklone*, 659 F.Supp. at 455, 465 (rejecting *Broderbund's* conclusion that audiovisual screen displays are copyrightable, although holding that separate copyright on status screen display was infringed where "total concept and feel" was copied); *Plains Cotton Cooperative Association v. Goodpasture Computer Service, Inc.*, 807 F.2d 1256, 1262 (rejecting *Whelan's* protection for structure, sequence, and organization, court instead held that sequence and organization, where dictated by market forces, is non-copyrightable idea rather than copyrightable expression), *reh'g denied*, 813 F.2d 407 (5th Cir.), *cert. denied*, 484 U.S. 821, 108 S.Ct. 80, 98 L.Ed.2d 42 (1987). In any event, *Synercom's* input formats are quite different from, and distinguishable from, the nonliteral aspects of 1-2-3 at issue in this case.

Defendant's proposed analogy is also similar to the analogy drawn by Commissioner Hersey between a computer program and an object that is designated to do work—for example, the cam of a drill. CONTU, *Final Report at* 58-60 (Hersey, C., dissenting). His view, however, was in dissent, and not a view advanced by CONTU. Because Congress adopted CONTU's recommendations practically verbatim, it is reasonable to infer that Congress did not adopt Commissioner Hersey's view.

Moreover, I conclude that defendants' contentions, to the extent they are similar to *Synercom's* central proposition and to Commissioner Hersey's views in dissent, are inconsistent

with the legislative history and statutory mandates explained above. If, in a context such as that of *Synercom* or of this case, an idea and its expression were taken to be inseparable and the expression therefore not copyrightable, copyright law never would, as a practical matter, provide computer programs with protection as substantial as Congress has mandated—protection designed to extend to original elements of expression however embodied. I credit the testimony of expert witnesses that the bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding, and that creating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine. *See* Galler Decl. ¶¶ 37, 39; Emery Decl. ¶¶ 22, 25-28 (Docket No. 265); Reed Decl. ¶¶ 26-27 (Docket No. 290). Defendants' contentions would attribute to the statute a purpose to protect only a narrowly defined segment of the creative development of computer programs, and to preclude from protection even more significant creative elements of the process. Such a result is fundamentally inconsistent with the statutory mandates.

Also, defendants' contention would have the additional consequence that computer programmers would have little, if any, more protection for nonliteral elements of expression embodied in their original works of authorship than is already provided by trade secret law. If the intellectual effort and creativity embodied in a user interface were protectable only by trade secret law, the length of protection for computer programs would be very short—merely the time it takes to examine a program and then duplicate the nonliteral elements in a newly written computer program. This short period of protection is fundamentally inconsistent with the mandates of the copyright law.

Defendants have advanced their "useful article" (or "functionality") argument in many forms. Never, however, have they stated every premise that is a step of the reasoning implicit in the argument. One set of unstated premises is the

assumed meanings of "useful," "article," and "useful article." Indeed, different forms of their argument have depended on different assumed meanings for one or more of these three terms. Some illustrations will clarify this point.

Suppose, first, the form of the argument is this:

> A "useful article" is not copyrightable; a "computer program" is an "article," and a good "computer program" is "useful"; therefore, a good "computer program" is not copyrightable.

In this illustration, "useful," "article," and "useful article" are assumed to have meanings consistent with ordinary usage (with an exception to be noted below). The proposition that emerges when the words are interpreted in this way is, of course, plainly contrary to congressional mandates. Congress must have had some purpose in defining "computer program" and declaring that an "owner of a copy of a computer program" may make additional copies, in specified circumstances, without violating copyright laws. 17 U.S.C. §§ 101, 117. The clear implication of the 1980 amendments is that in some circumstances, at least, a good computer program *is* copyrightable. Otherwise, there would have been no need for Congress to enact these two provisions about a "computer program" and additional "copies."

It follows that, for this defense argument to make any sense and also be consistent with congressional mandates, defendants must be proposing that "useful", or "article," or "useful article," or all three, be interpreted in some sense different from their ordinary meanings. In what sense, though, they never say.

Suppose, second, a different form of the argument:

> A "useful article" is not copyrightable; a "screen display" is an "article," and a good "screen display" is "useful"; therefore, a good "screen display" is not copyrightable.

Here, the implicit meaning of "article" is being stretched beyond ordinary usage to apply to what one would ordinarily think of as just one part of an article, rather than itself an arti-

cle. Also, the implicit meaning of "article" stretches beyond what one would ordinarily think of as something that, as part of a larger "article," is itself an "article." Thus, the implicit meaning of "article" is broader still than the statutory mandate that "[a]n *article* that is normally a part of a useful article is considered a 'useful article.' " 17 U.S.C. § 101 (emphasis added). Are these expansions of the meaning of "article" consistent with the text and with the object and policy manifested in the statute? We may be better prepared to answer after considering one more illustrative form of the argument.

Suppose, third, this form of the argument:

> A "useful article" is not copyrightable; a "user interface" is an "article," and a good "user interface" is "useful"; therefore, a good "user interface" is not copyrightable.

Here, the meaning of "article" is plainly stretched beyond ordinary usage, and especially when we understand "user interface" in a sense broad enough to include nonliteral elements such as the command structure. Moreover, unease that we are being led into a departure from statutory text and manifested object and policy grows deeper as we reflect still more on this illustrative form of the argument.

One problem is that the argument depends on changing the assumed meaning of one or all of "useful," "article," and "useful article," to suit the needs of the copier who is advancing the argument as a defense against a claim of copyright infringement.

An even deeper fallacy of the argument, however, is that it assumes a meaning of "useful article" in step one ("a 'useful article' is not copyrightable") that goes far beyond ordinary meanings of "useful" and "article" combined. That is, not merely does it assume a *descriptive* meaning of something that is an "article" and is also "useful," but in addition it assumes that the definition of "useful article" includes, as something built into it, a rule of law: Everything that is a "useful article" in a descriptive sense is also, by rule of law, not copyrightable, and under this rule of law, nothing about a

"useful article"—no element, no aspect, no part of it—can ever be copyrighted. Merely to expose this assumed prescriptive meaning of "useful article," as defendants persistently use the phrase, is to demonstrate that any argument founded upon this meaning is fundamentally inconsistent with the mandates of the copyright statute.

It may be quite true, with respect to "useful articles"—indeed I believe it to be so—that their utilitarian aspects are not copyrightable, and that things that *merely* utter work, such as the cam of a drill, are not copyrightable. It is not true, however, that every aspect of a user interface that is "useful" is therefore not copyrightable. For example, Lotus 1-2-3 is surely "useful." It does not follow that when an intellectual work achieves the feat of being useful as well as expressive and original, the moment of creative triumph is also a moment of devastating financial loss—because the triumph destroys copyrightability of all expressive elements that would have been protected if only they had not contributed so much to the public interest by helping to make some article useful.

Defendants' contention misses this point by proceeding on an erroneous assumption about the role of "functionality" in copyright law. It is true that "functionality" of an article does not itself support copyrightability. Thus, it never strengthens a claim for copyright to show that the "work" for which copyright protection is claimed is useful. A congressional mandate that "proof that an intellectual work is a 'useful article' does not support the author's claim for copyright" is not, however, a mandate that "if one who copied the author's work proves that the work was 'useful' or 'functional,' the author loses *all* copyright protection." Transforming a mandate that "proof of usefulness *does not strengthen* a copyright claim" to a mandate that "proof of usefulness *destroys* a copyright claim" is, to say the least, a remarkable intellectual leap. Defendants have not advanced such a proposition explicitly. But this is in fact an implicit premise of their contention—or a consequence of it, if one takes a hindsight view of having applied their proposed rule in decisionmaking. In effect, their proposed rule would work this way: Anything that is useful is a "use-

ful article"; nothing about a "useful article" is ever copyrightable; because 1-2-3 is useful, and is an article, it is not copyrightable.

A more sensible interpretation of the statutory mandate is that the mere fact that an intellectual work is useful or functional—be it a dictionary, directory, map, book of meaningless code words, or computer program—does not mean that none of the elements of the work can be copyrightable. Also, the statute does not bar copyrightability merely because the originality of the expression becomes associated, in the marketplace, with usefulness of the work to a degree and in dimensions not previously achieved by other products on the market. *Brandir International, Inc. v. Cascade Pacific Lumber Co.*, 834 F.2d 1142, 1147 (2d Cir.1987) ("[A] copyrighted work . . . does not lose its protected status merely because it subsequently is put to functional use."); *NEC Corp. v. Intel Corp.*, 645 F.Supp. 590, 595 (N.D.Cal.1986) ("function performed by defendant's microprograms . . . does not affect their status as copyrightable subject matter"), *vacated on grounds of judge's recusal, see* 835 F.2d 1546 (9th Cir.1988). To hold otherwise would be to deny copyright protection to the most original and least obvious products of the creative mind merely because the marketplace accepts them as distinctively "functional." Such a rule would grant copyright protection for only those products that fall far short of being the best available. Rather than promoting and encouraging both the development and disclosure of the best, such a rule would offer incentives to market only the second, or third or tenth best, and hold back the best for fear that it is too good for copyrightability. Copyrightability is not a synonym for imperfection.

Accordingly, I conclude that a court, in determining whether a particular element is copyrightable, must not allow one statutory mandate—that functionality or usefulness is not itself a basis for copyrightability—to absorb and destroy another statutory mandate—that elements of expression are copyrightable. Elements of expression, even if embodied in useful articles, are copyrightable if capable of identification

and recognition independently of the functional ideas that make the article useful. This mandate may be viewed as a corollary of the central distinction of copyright law between idea and expression, which is explored further immediately below.

B. The Idea-Expression Riddle: Four Additional Concepts

It is by now plain that an idea is not copyrightable and an expression may be. It does not follow, though, that every expression of an idea is copyrightable. To begin to get an understanding of the legally significant contrasts among an idea, noncopyrightable expressions of the idea, and a copyrightable expression, we must take account of four more concepts.

Earlier parts of this Opinion refer to two of these four—"originality" and "functionality." The expression of an idea is copyrightable only if it is original—that is, if the expression originated with the author. 17 U.S.C. § 102(a); *see* Part III(A)(2)(b), *supra*. Even then the expression of the idea is not copyrightable if the expression does no more than embody elements of the idea that are functional in the utilitarian sense. 17 U.S.C. § 102(b); *see* Part IV(A) *supra*.

The third concept is "obviousness." When a particular expression goes no farther than the obvious, it is inseparable from the idea itself. Protecting an expression of this limited kind would effectively amount to protection of the idea, a result inconsistent which the plain meaning of the statute. *E.H. Tate Co. v. Jiffy Enterprises, Inc.*, 16 F.R.D. 571, 573 (E.D.Pa.1954) (small sketch and accompanying instruction "Apply hook to wall" so obvious that it is not entitled to copyright protection).

It is only a slight extension of the idea of "obviousness"—and one supported by precedent—to reach the fourth concept: "merger." If a particular expression is one of a quite limited number of the possible ways of expressing an idea, then, under this fourth concept, the expression is not copyrightable:

> When the uncopyrightable subject matter is very narrow, so that "the topic necessarily requires," if not only one

> form of expression, at best only a limited number, to permit copyrighting would mean that a party or parties, by copyrighting a mere handful of forms, could exhaust all possibility of future use of the substance. In such circumstances it does not seem accurate to say that any particular form of expression comes from the subject matter. However, it is necessary to say that the subject matter would be appropriated by permitting the copyrighting of its expression. We cannot recognize copyright as a game of chess in which the public can be checkmated.

Morissey v. Procter & Gamble Co., 379 F.2d 675, 678-79 (1st Cir.1967) (citations omitted). *See also Concrete Machinery Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir.1988) ("When there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression."); *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir.1971) (idea of a jewel-encrusted life-like bee pin inseparable from expression; thus expression not copyrightable because "protecting the 'expression' in such circumstances would confer a monopoly of the 'idea' upon the copyright owner"). *Cf Atari, Inc. v. North American Philips Consumer Electronics Corp*, 672 F.2d 607, 616 (7th Cir.), *cert. denied*, 459 U.S. 880, 103 S.Ct. 176, 74 L.Ed.2d 145 (1982) (*scènes à faire* of literary works—"stock literary devices" such as "incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic"—"are not protectible [sic] by copyright"); *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485, 489 (9th Cir.) (*scènes à faire* not copyrightable because granting a copyright "would give the first author a monopoly on the commonplace ideas behind the *scènes à faire*"), *cert. denied*, 469 U.S. 1037, 105 S.Ct. 513, 83 L.Ed.2d 403 (1984).

If, however, the expression of an idea has elements that go beyond all functional elements of the idea itself, and beyond the obvious, and if there are numerous other ways of express-

ing the non-copyrightable idea, then those elements of expression, if original and substantial, are copyrightable.

C. Elements of the Legal Test for Copyrightability

A "legal test," as I use the phrase here, is a statement of the elements of fact, or law, or both fact and law, that must be addressed by a decisionmaker to decide a question potentially decisive of some claim or defense. "Copyrightability" of non-literal elements of Lotus 1-2-3 is essential to the claim of the plaintiff in this case.

As already noted, the legal test for deciding copyrightability, in a factual context such as is presented here, has not been precisely defined either in the copyright statute or in precedents interpreting and applying it. Nevertheless, the statute and the precedents contain many mandates—"markers" of the borderline between copyrightability and non-copyrightability—that narrow the scope of the questions remaining to be answered to determine what test to apply.

Drawing into one statement the fundamental truths about ideas and their expression that were sketched above, one may accurately say that the issue of copyrightability of a "work" turns not on whether the work expresses ideas but instead on whether, in addition to expressing one or more ideas, in some material respect it does more, and in an original way. One need not totally disentangle the idea from its expression in order to conclude that a particular aspect is expression. Indeed, to speak as if it were ever possible completely to disentangle an idea from an expression of that idea is to speak abstract fiction rather than real-life fact. Disentanglement, then, is not an "either-or," "0-1," "negative-positive," or "binary" matter. It is, instead, a matter of degree.

Still, even if the "idea" cannot be completely disentangled from its expression, to determine what is copyrightable a decisionmaker must understand the meaning of "idea" within the idea-expression distinction. To do so one must take note also of another distinction—one between generality and specificity of conceptualizing the idea. Thus, a statement of

the most significant elements of the legal test for copyrightability, consistent with precedents, begins:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, *along the scale from the most generalized conception to the most particularized*, and choose some formulation—some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression.

As Learned Hand recognized in a 1930 case concerning the alleged infringement of the copyright of a play:

> Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended. Nobody has ever been able to fix that boundary, and nobody ever can.

Nichols, 45 F.2d at 121 (citations omitted). *See also Shipman v. RKO, Radio Pictures*, 100 F.2d 533, 538 (2d Cir.1938) (L. Hand, J., concurring) ("*Nichols* . . . held that there is a point where the similarities are so little concrete (and therefore so abstract) that they become only 'theme', 'idea', or skeleton of the plot, and that these are always in the public domain; no copyright can protect them. The test is necessarily vague and nothing more definite can be said about it.") Thirty more years of experience in judging did not change Learned Hand's view: "Obviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression.' Decisions must therefore inevitably be ad hoc." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir.1960) (L. Hand, J.). In another context,

Hand described such "ad hoc" decisionmaking as "fiat." *Sinram v. Pennsylvania Railroad Co.*, 61 F.2d 767, 771 (2d Cir.1932) (L. Hand, J.). In whatever way this kind of decisionmaking may be described, Hand offered us no formula for distinguishing between idea and expression like that he devised for the calculus of reasonable care in *United States v. Carroll Towing Co.*, 159 F.2d 169, 173 (L. Hand, J.) (articulating the "BPL" formula), *reh'g denied*, 160 F.2d 482 (2d Cir.1947). It seems the better part of wisdom, if not valor, not to press the search for a suitable brightline test of copyrightability where Learned Hand, even after decades of experience in judging, found none.

For all these reasons, as a practical necessity, whether explicitly or only implicitly, courts apply an abstractions scale in determining copyrightability. In doing so they make a decision involving choice and judgment of a type that human minds make regularly in daily affairs, but computers of the current state of the art cannot make.[2] Still, though "judgment" is required and the answer to be given is not precisely "calculable," analogies to arithmetic calculations and to "scales of justice" may aid the human mind in choosing and "weighing" factors that properly go into the metaphoric calculus.

In summary, one among the principal elements to be weighed in determining copyrightability when the idea-expression distinction applies is to conceive and define the idea in a way that places it somewhere along the scale of abstraction (somewhere between the most abstract and the most specific of all possible conceptions). Illustrations from the evidence in this case will help to explain in a more concrete way this element of the legal test for copyrightability. *See* Part V(C), *infra*. Before turning to those illustrations, however, I state in a similarly abstract way, to be explained

---

[2] Despite *Time* Magazine's decision to honor "the computer" as its 1982 "Man of the Year," and despite advances in the field of artificial intelligence, I take it as a premise of decisionmaking in this case that computers, for better or worse, do not yet have the human feelings, strengths, and failings that lie beyond those we describe as logical, or cognitive, or intellectual in the broadest sense.

later by illustrations, two more elements that I conclude a decisionmaker must consider to determine copyrightability of a computer program like that at issue in this case.

In addition to taking account of the distinction between generality and specificity, to make use of Hand's abstraction scale for applying the idea-expression distinction we need to identify and distinguish between essential and nonessential details of expressing the idea. Some, but of course not all, details, are so essential that their omission would result in a failure to express *that* idea, or in the expression of only a different and *more general* idea. Accordingly, two more elements in the legal test for copyrightability are:

> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea.
>
> THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."

In addressing this third element of the test for copyrightability, the decisionmaker is measuring "substantiality" not merely on a quantum scale but by a test that is qualitative as well. *SAS Institute*, 605 F.Supp. at 829-30 ("the piracy of even a quantitatively small fragment ('a rose by any other name would smell as sweet') may be qualitatively substantial").

By its nature, a legal test that requires weighing of factors or elements such as these is not a bright-line or an either-or test. It requires of the decisionmaker, instead, an evaluative or "judgmental" weighing of all relevant characteristics of the work in which a copyright is claimed, all relevant characteristics of the allegedly infringing work, and all of the relevant circumstances of their development and use. It requires, also, not a step-by-step decisionmaking process, but a simultaneous weighing of all the factors or elements that the legal

test identifies as relevant. I do not suggest that the three elements identified here are an all-inclusive list. They do appear, however, to be the principal factors relevant to decision of copyrightability of a computer program such as Lotus 1-2-3.

If the decisionmaker, weighing the relevant factors, determines that the legal test applying the idea-expression distinction is satisfied, copyrightability is established. Issues may remain, of course, as to whether a copyright was perfected and whether the alleged infringing work, measured by the "substantial similarly" test, did contain elements that infringed upon the copyrightable elements of the copyrighted "work." Also, issues may remain as to whether damages have been proved, or whether the controversy over infringement is instead " 'a trivial pother,' a mere point of honor, of scarcely more than irritation, involving no substantial interest," *Fred Fisher, Inc. v. Dillingham*, 298 F. 145, 152 (S.D.N.Y.1924) (L. Hand, J.) (citations omitted), for which only statutory damages should be awarded. 17 U.S.C. § 504(c) (1988).

D. Incentives and the Role of Advocacy

The advocate of broad copyrightability who understands that the court will apply a legal test that focuses principally on these three elements has an incentive to urge that the court conceive the "idea" in a very generalized sense; then many different expressions of the idea would be possible, and protection might be claimed for the work. The advocate of freedom to copy—and of narrow copyrightability—has an incentive to urge that the court conceive the "idea" in a very particularized sense; then only one or a few expressions of an idea defined in such particularized terms would be possible, and no copyright protection for those few expressions would be available because the idea and expression would merge completely, or nearly so. Such extreme positions would fail to assist the court in determining, for purposes of the first element, where properly to place the idea along the abstractions scale.

There is risk for each advocate, however, in yielding too readily to these respective incentives. The argument of an

advocate who presses too far in one or the other of these directions of generality or specificity in defining the idea will not only lose the argument advanced but also lose credibility for later advancing a more sensible alternative that proposes a less extreme but still favorable position along the scale. Upon reflection, then, advocates on both sides will be—or at least should be—encouraged to moderate their ultra-contentious and extreme positions in favor of more supportable propositions that will more sharply focus the issue for adjudication.

There is an additional risk for the advocate of freedom to copy who attempts to persuade the court that Hand's abstractions scale, and the three-element test applying it, is irrelevant to computer programs. That advocate will be tempted to define the idea in the most generalized terms, hoping to persuade the court that, because the abstractions scale does not apply, all of the many different expressions of that idea are non-copyrightable and everyone is free to copy.

In this case, defendants have in fact advanced such an argument. Yielding only to the near-uniformity of precedent for copyrightability of source code and object code, defendants have argued that every other form of expression of the "spreadsheet metaphor" is non-copyrightable. For reasons now to be explained, neither this extreme contention nor any of the alternatives the defendants have advanced for non-copyrightability can be sustained.

V. APPLICATION OF THE LEGAL TEST TO LOTUS 1-2-3

A. *"Look and Feel"*

In musical, dramatic, and motion picture works, and works of literature, nonliteral elements that are copyrightable have sometimes been described as the "total concept and feel" of a work, *Roth Greeting Cards*, 429 F.2d at 1110; *Krofft Television*, 562 F.2d at 1167, "the fundamental essence or structure" of a work, 3 M. Nimmer & D. Nimmer, *Nimmer on Copyright* § 13.03[A][1] (1989), or "the 'pattern' of the

work," Chaffee, *Reflections on the Law of Copyright: Part I*, 45 Colum.L.Rev. 503, 513 (1945). In the context of computer programs, nonliteral elements have often been referred to as the 'look and feel" of a program, *e.g.*, *Telemarketing Resources*, 12 U.S.P.Q.2d at 1993; Samuelson and Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurimetrics 121 (1989). Initially, plaintiff too referred to these elements as "look and feel," Amended Complaint ¶ 13 (Docket No. 15), though plaintiff—in trial, at least—has not rested its contentions primarily on this terminology.

Despite its widespread use in public discourse on the copyrightability of nonliteral elements of computer programs, I have not found the "look and feel" concept, standing alone, to be significantly helpful in distinguishing between nonliteral elements of a computer program that are copyrightable and those that are not.

One may argue that the phrase "look and feel" is analogous to the "total concept and feel" test developed in *Roth Greeting Cards*, 429 F.2d at 1110, and used in *Krofft Television*, 562 F.2d at 1167. In these cases, however, the "total concept and feel" test was not invoked—at least, not explicitly—as an aid to the court in determining which nonliteral elements were copyrightable and why. Rather, these courts used the concept, not in determining copyrightability, but, apparently assuming copyrightability, in applying the substantial similarity test to determine whether forbidden copying had occurred. For example, in *Roth Greeting Cards*, the court considered whether the "text, arrangement of text, art work, and association between art work and text" of defendant's greeting cards were substantially similar to (*e.g.*, copied the "total concept and feel" of) those elements of plaintiff's greeting cards. 429 F.2d at 1109. And in *Krofft Television*, the inquiry focused on whether defendant McDonald's, with its television commercials, copied the "total concept and feel" of plaintiff's H.R. Pufnstuf children's television series by copying its locale (both occurred in imaginary world with similar trees, caves, a pond, a road and a castle), fictional characters (both were

inhabited by talking trees, mayors with disproportionately large round heads and long wide mouths, "Keystone cops" characters, crazy scientists, and a multi-armed evil creature), costumes and sets (created for defendant by former employees of plaintiff), voices (executed for defendant by same voice expert who did the voices for plaintiff's television series), and plot. 562 F.2d at 1161, 1167 n. 9.

It may be true that the issues of copyrightability and substantial similarity are so interrelated that these precedents are relevant. The fact remains that they are not directly in point for determining copyrightability in this case. Moreover, "look and feel" is a conclusion, and the usefulness and applicability of a precedent depends on the *reasons* the conclusion was reached in a particular context, not on the *conclusion* itself. Thus, in trying to understand the relevance of "concept and feel" precedents, we need to look to details of those cases that appear to have been relied upon in reaching the conclusion, rather than merely embracing the conclusion without regard for underlying reasons. As we probe the circumstances of these precedents that closely, we are likely to do something akin to apply in the three-element test described above.

B. The User Interface

Plaintiff in the present case, not now pressing any argument that the phrase "look and feel" is a satisfactory description of the copyrightable elements of Lotus 1-2-3, suggests that the copyrightable nonliteral elements are more appropriately described by the phrase "user interface." According to plaintiff, the "user interface" of 1-2-3 includes such elements as "the menus (and their structure and organization), the long prompts, the screens on which they appear, the function key assignments, [and] the macro commands and language." Plaintiff's Post-Trial Brief at 53 (Docket No. 319). I turn now to examining these elements more closely.

Like manual spreadsheets, *Baker*, 101 U.S. at 100, the electronic spreadsheet presents a blank form on which numerical, statistical, financial or other data can be assimilated, organized, manipulated and calculated. Galler Decl. ¶¶ 101-103;

Lewis Affdvt. ¶ 92. In both Lotus 1-2-3 and VP-Planner, as in many other electronic spreadsheet programs, a highlighted element of the basic screen display resembles an "L" rotated ninety degrees clockwise with letters across the top to designate columns, and numbers down the left side to designate rows. *See* Appendices 1 (VisiCalc), 2 (1-2-3), and 3 (VP-Planner). *Cf.* Multiplan, which also has a rotated "L" screen display, but which uses numbers for both columns and rows (Tr. Ex. (Trial Exhibit) 132). The intersection of each column and row is a "cell" in which a value (*e.g.*, 31,963), formula (*e.g.*, one that adds a column of numbers), or label (*e.g.*, "Cost of Goods") may be entered.

Both programs utilize a "two-line moving-cursor menu," which presents the user with a list of command choices (*e.g.*, "file", "copy", "quit") and a moving cursor to use in communicating ("entering") the choice. The menu is called up to the screen by pressing the slash ("/") key, and is located either above the rotated "L" (as in 1-2-3, *see* Appendix 2) or below the rotated "L" as in VP-Planner, *see* Appendix 3). *Cf.* Multiplan, which uses a three-line moving cursor menu (Tr. Ex. 132); Excel, which has "pull-down" bar menus (Tr. Ex. 79).

The top line of the two-line menu contains a series of words, each of which represents a different command. For example, the top line of the first, or main, menu in 1-2-3 reads: "Worksheet Range Copy Move File Graph Data Quit". *See* Appendix 2; *cf.* Appendix 3 (main menu in VP-Planner). The first word of the line is highlighted to signify the command that will be chosen if the "enter" key is pressed; the highlighting, or "cursor," moves to the right or left if the right or left cursor key is pressed.

The second line of the menu displays a "long prompt," which contains further information about the highlighted command. In some cases, the long prompt is a description of the highlighted command (*e.g.*, for command "Copy", the long prompt reads: "Copy a cell or range of cells"); in other cases, the long prompt provides a list of the menu command subchoices that will be available if the highlighted command is chosen (*e.g.*, for command "Worksheet", the long prompt

reads: "Global, Insert, Delete, Column-Width, Erase, Titles, Window, Status"; *see* Appendix 2; *cf.* Appendix 3 (VP-Planner)). In the latter case, if the highlighted command is subsequently chosen, the words that appeared in the long prompt will now appear as second-level menu command choices on the top line of the menu, and a new long prompt will take its place on the second line.

In addition to having the option of selecting a command by moving the cursor to the command and pressing the "enter" key, a user may instead press the key representing the first letter of the command word (*e.g.*, "C" for "Copy", "W" for "Worksheet"). For this reason, each word representing each command on a given menu line must start with a different letter. *See* Galler Affdvt. ¶¶ 105-117; Lewis Decl. ¶¶ 96-97.

Function keys present an additional way for the user to communicate with and operate, the programmed computer. Each program assigns certain frequently-used commands to the various 'function keys" labelled "F1", "F2", "F3", etc.) on the keyboard. For example, in 1-2-3, "F1" corresponds to the command "Help", and "F2" to the command "Edit". Galler Decl. ¶ 125. VP-Planner, in contrast, assigns the function keys to the commands on the top line of the menu. Thus, for example, when VP-Planner is in its main menu mode, "F1" corresponds to the command "Help", "F2" to the command "Worksheet", "F3" to the command "Range", "F4" to the command "Copy", and so on. *See* Appendix 3 (function key numbers listed before command terms).

Typically, users adapt particular spreadsheets to their specific needs. Suppose, for example, that in order to achieve a desired result, a user must perform the same sequence of commands repeatedly in order to cause the computer to execute the same functions repeatedly (*e.g.*, calculating depreciation based on certain financial data, or saving the spreadsheet and printing a copy of it). Rather than going step-by-step through the same sequence of commands each time there is a need to perform a particular function, the user may store a sequence of command terms as a "macroinstruction," commonly called a "macro," and then, with one command stroke that invokes

the macro, cause the programmed computer to execute the entire sequence of commands.

In 1-2-3, the command terms within a macro may consist of menu choices (*e.g.*, "/C" to copy a cell or range of cells, or "/PPRA1.F19{enter}AGQ"—for the command sequence "Print, Printer, Range, Cell A1 to Cell F19, Align, Go, Quit"—to print the specified range of cells from the spreadsheet). Also a macro may include keyboard commands (such as function keys, cursor keys, or the "enter" key), and special macro commands invoked by the command "/X" (*e.g.*, "XI" is a macro command that performs an "if-then" logical function). Because macros may contain many menu choices, the exact hierarchy—or structure, sequence and organization—of the menu system is a fundamental part of the functionality of the macros. Also, because macro commands are typically invoked by entering the first letter of command choices (*e.g.*, "/C" for copying, "/PPRA1.F19{enter}AGQ" for printing the specified range of cells), the first letter of each command choice on a particular menu is a vital element of the functionality of macros. *See* Galler Decl. ¶¶ 121-124; Lewis Affdvt. ¶¶ 109-115.

C. Elements of the User Interface as Expression

Applying to 1-2-3 the legal test stated in Part IV(C), *supra*, I consider first where along the scale of abstraction to conceive the "idea" for the purpose of distinguishing between the idea and its expression.

At the most general level of Hand's abstractions scale, *Nichols*, 45 F.2d at 121—the computer programs at issue in this case, and other computer programs that have been considered during the course of trial, are expressions of the idea of a computer program for an electronic spreadsheet. Defendants are quite correct, then, in asserting that the idea of developing an electronic spreadsheet is not copyrightable—that the core idea of such a spreadsheet is both functional and obvious, even to computer users who claim no technical competence. Thus, even though programs like VisiCalc, 1-2-3, Multiplan, SuperCalc4, and Excel are very different in their

structure, appearance, and method of operation, each is, at the most basic level, just a different way of expressing the same idea: the electronic spreadsheet. It does not follow, however, that every possible method of designing a metaphorical spreadsheet is obvious, or that no form of expressing the idea of the spreadsheet metaphor can possibly have such originality in pressing beyond the obvious as is required for copyrightability, or that no special form of metaphorical spreadsheet can possibly be a distinctive expression of a particular method of preparing financial information.

The idea for a two-line moving cursor menu is also functional and obvious, and indeed, is used in a wide variety of computer programs including spreadsheet programs. Nevertheless, it does not follow that every possible method of designing a menu system that includes a two-line moving cursor is non-copyrightable.

Of course, if a particular expression of the idea of an electronic spreadsheet communicates no details beyond those essential to stating the idea itself, then that expression would not be copyrightable. The issue here is whether Lotus 1-2-3 does go beyond those details essential to any expression of the idea, and includes substantial elements of expression, distinctive and original, which are thus copyrightable.

The idea for an electronic spreadsheet was first rendered into commercial practice by Daniel Bricklin. As a student at Harvard Business School in the late 1970's, Bricklin envisioned a "magic blackboard" that would recalculate numbers automatically as changes were made in other parts of the spreadsheet. Eventually, aided by others, he transformed this idea into VisiCalc, the first commercial electronic spreadsheet. *See* Bricklin Affdvt. ¶ 48-96 (Docket No. 217). Bricklin's idea for VisiCalc was a revolutionary advance in the field of computer programming. Dauphinais Affdvt. ¶ 98 (Docket No. 280).

Although VisiCalc was a commercial success, implementational characteristics limited the scope and duration of its marketability as a spreadsheet product. Most notably, VisiCalc was originally programmed for use on the Apple II computer, which had limited memory (32K of RAM), limited

screen display capabilities (only 40 characters per line), and limited keys available on the keyboard (no function keys and no up and down cursor keys). When VisiCalc was later rewritten for use on the IBM PC which was introduced in August 1981), it was transferred with minimal changes and without taking advantage of many of the PC's more extensive capabilities.

Mitchell Kapor and Jonathan Sachs, the original authors of 1-2-3, exploited this opportunity. Building on Bricklin's revolutionary idea for an electronic spreadsheet, Kapor and Sachs expressed that idea in a different, more powerful way. 1-2-3 took advantage of the IBM PC's more expansive memory and more versatile screen display capabilities and keyboard. 1-2-3, like many electronic spreadsheet programs since, could thus be thought of as an evolutionary product that was built upon the shoulders of VisiCalc.

Just as 1-2-3 expressed the idea of an electronic spreadsheet differently from VisiCalc, so did Microsoft's Excel. Originally written for the Apple Macintosh computer, it exploits the enhanced graphics capabilities of the Macintosh, as well as the mouse input device that is standard with the Macintosh. Excel has pull-down bar menus rather than a two-line moving-cursor menu, and a very different menu-command hierarchy. Tr. Ex. 79.

As already noted, these three products—VisiCalc, 1-2-3, and Excel—share the general idea of an electronic spreadsheet but have expressed the idea in substantially different ways. These products also share some elements, however, at a somewhat more detailed or specific point along the abstractions scale. One element shared by these and many other programs is the basic spreadsheet screen display that resembles a rotated "L." *See* Appendices 1 (VisiCalc), 2 (1-2-3), and 3 (VP-Planner). Although Excel uses a different basic spreadsheet screen display that more closely resembles a paper spreadsheet, there is a rather low limit, as a factual matter, on the number of ways of making a computer screen resemble a spreadsheet. Accordingly, this aspect of electronic spreadsheet computer programs, if not present in every expression of such a program, is present in most expressions. Thus the

second element of the legal test weighs heavily against treating the rotated "L" screen display as a copyrightable element of a computer program. *Morrissey*, 379 F.2d at 678-79

Another expressive element that merges with the idea of an electronic spreadsheet—that is, that is an essential detail present in most if not all expressions of an electronic spreadsheet—is the designation of a particular key that, when pressed, will invoke the menu command system. The number of keys available for this designation is limited for two reasons. First, because most of the keys on the keyboard relate either to values (*e.g.*, the number keys and mathematical operation keys) or labels (*e.g.*, the letter keys), only a few keys are left that can be used, as a practical matter, to invoke the menu command system. Without something more, the programmed computer would interpret the activation of one of these keys as an attempt by the user to enter a value or label into a cell. Second, because users need to invoke the command system frequently, the key designated for this purpose must be easily accessible. For example, the user should not be required to press two keys at the same time (such as "Shift," "Alt," or "Ctrl" along with another key).

As just noted, when all the letter, number, and arithmetic keys are eliminated from consideration, the number of keys remaining that could be used to invoke the menu command system is quite limited. They include the slash key ("/") and the semi-colon key (";"). The choice of the creators of VisiCalc to designate the slash ("/") key to invoke the menu command system is not surprising. It is one of very few practical options. Thus the second element of the legal test weighs heavily against copyrightability of this aspect of VisiCalc—and of 1-2-3. This expression merges with the idea of having a readily available method of invoking the menu command system.

Other elements of expression a decisionmaker may regard as either essential to every expression of an electronic spreadsheet, or at least "obvious" if not essential, include the use of the "+" key to indicate addition, the "—" key to indicate subtraction, the "*" key to indicate multiplication, the "/" key within formulas to indicate division, and the "enter" key to place keystroke entries into the cells. *See* Dauphinais Affidavit, ¶ 78.

Each of the elements just described is present in, if not all, at least most expressions of an electronic spreadsheet computer program. Other aspects of these programs, however, need not be present in every expression of an electronic spreadsheet. An example of distinctive details of expression is the precise "structure, sequence, and organization," *Whelan*, 797 F.2d at 1248, of the menu command system.

Consider first the menu command system of VisiCalc. The main menu command line reads: "Command: BCDEFGIMPRSTVW-". *See* Appendix 1. Each of these letters (or, to use defendants' experts' preferred terminology, "symbolic tokens") stands for a different command—in this case: Blank, Clear, Delete, Edit, Format, Global, Insert, Move, Print, Replicate, Storage, Titles, Version Number, Window, and "-" for "Label Repeating." Many of these commands invoke submenus which also contain a series of letters, each of which represents a submenu command choice. *See* VisiCalc Command Structure Chart (Tr. Ex. 140, pp. 3-3 and 3-4).

This particular expression of a menu structure is not essential to the electronic spreadsheet idea, nor does it merge with the somewhat less abstract idea of a menu structure for an electronic spreadsheet. The idea of a menu structure—including the overall structure, the order of commands in each menu line, the choice of letters, words, or "symbolic tokens" to represent each command, the presentation of these symbolic tokens on the screen (*i.e.*, first letter only, abbreviations, full words, full words with one or more letters capitalized or underlined), the type of menu system used (*i.e.*, one-, two-, or three-line moving-cursor menus, pull-down menus, or command-driven interfaces), and the long prompts—could be expressed in a great many if not literally unlimited number of ways.

The fact that some of these specific command terms are quite obvious or merge with the idea of such a particular command term does not preclude copyrightability for the command structure taken as a whole. If particular characteristics

not distinctive individually have been brought together in a way that makes the "whole" a distinctive expression of an idea—one of many possible ways of expressing it—then the "whole" may be copyrightable. The statutory provisions regarding "compilation," 17 U.S.C. §§ 101, 103, are not essential to this conclusion, but do reinforce it. A different total structure may be developed even from individual components that are quite similar and limited in number. To determine copyrightability, a court need not—and, indeed, should not—dissect every element of the allegedly protected work. Rather, the court need only identify those elements that are copyrightable, and then determine whether those elements, *considered as a whole*, have been impermissibly copied. *Atari Games Corp. v. Oman*, 888 F.2d 878, 882-83 (D.C.Cir. 1989) (rejecting "component-by-component analysis," and ruling instead that focus must ultimately be on "work as a whole").

It is plain that plaintiff did not impermissibly copy copyrighted elements of VisiCalc. Lotus 1-2-3 uses a very different menu structure. In contrast with VisiCalc's one-line main menu that reads "Command: BCDEFGIMPRSTVW-", the main menu of Lotus 1-2-3, which uses a two-line moving-cursor menu system, reads: "Worksheet Range Copy Move File Graph Data Quit". *See* Appendix 2. Most of the submenus similarly present a list of up to about ten full-word menu choices, presented in order of predicted frequency of use rather than alphabetically. *See* Lotus 1-2-3 Command Tree Chart (Tr. Ex. 176, pp. 3-2 and 3-3). Other spreadsheet programs have also expressed their command structures in completely different ways. *Cf.* Multiplan (Tr. Ex. 132, p. 279-374), Framework II (Tr. Ex. 131, pp. QR7-QR8), SuperCalc4 (Tr. Ex. 133, Quick Reference Card Number 1), Excel (Tr. Ex. 79, Reference Guide, p. 90); Symphony (Tr. Ex. 130, Reference Manual, pp. 127-190); MathPlan (Tr. Ex. 134, pp. 169-380); and PFS: Professional Plan (Tr. Ex. 135).

I conclude that a menu command structure is capable of being expressed in many if not an unlimited number of ways, and that the command structure of 1-2-3 is an original and nonobvious way of expressing a command structure. Emery

Decl. ¶ 15. Accordingly, the menu structure, taken as a whole—including the choice of command terms, the structure and order of those terms, their presentation on the screen, and the long prompts—is an aspect of 1-2-3 that is not present in every expression of an electronic spreadsheet. It meets the requirements of the second element of the legal test for copyrightability.

Finally, I consider the third element of the legal test—whether the structure, sequence, and organization of the menu command system is a substantial part of the alleged copyrighted work—here Lotus 1-2-3. That the answer to this question is "yes" is incontrovertible. The user interface of 1-2-3 is its most unique element, and is the aspect that has made 1-2-3 so popular. That defendants went to such trouble to copy that element is a testament to its substantiality. Accordingly, evaluation of the third element of the legal test weighs heavily in favor of Lotus.

Taking account of all three elements of the legal test, I determine that copyrightability of the user interface of 1-2-3 is established.

VI. COPYING OF LOTUS 1-2-3

As noted at the beginning of this Opinion, the parties' stipulation regulating this first phase of trial reserved for a later, jury phase, any issues of fact requiring jury deterrnination with respect to any alleged copying of Lotus 1-2-3. If in this first phase the court had rejected Lotus' claim of copyrightability of nonliteral elements of the user interface, a jury phase would plainly have been required under that stipulation to determine whether any copying of source code or object code had occurred. Because, however, the court has decided instead that Lotus prevails on this issue, the court must next consider whether any issue remains that must be submitted to a jury before the court considers whether defendants are liable for tnfringement of Lotus' copyright in 1-2-3.

For the reasons stated below, the answer to that inquiry must be "no." Not only is the copying in this case so "overwhelming and pervasive" as to preclude, as a matter of law, any assertion of independent creation, *see Midway Manufacturing Co. v. Bandai-America, Inc.*, 546 F.Supp. 125, 141 n. 11, 149 (D.N.J.1982) ("overwhelming and pervasive" copying can preclude, as a matter of law, finding of independent creation, and can support grant of summary judgment for plaintiff, but such virtual identity not shown in this case), but also, defendants in this case have *admitted* that they copied these elements of protected expression.

Dr. James Stephenson, founder of defendant Stephenson Software, is the original developer of the program that was eventually released as VP-Planner. Like Kapor and Sacks, Stephenson recognized that VisiCalc, although a "pioneering spreadsheet approach," Stephenson Affdvt. ¶ 19 (Docket No. 287), was not sufficient to meet the financial planning needs of some companies and did not take advantage of technological advances in computer hardware. Accordingly, in January 1982, Stephenson began to develop his own electronic spreadsheet that he referred to as FIPS ("Financial Information and Planning System"). *See id.* at ¶¶ 13-59.

By January 1983, Stephenson had developed much of the user interface for his spreadsheet program including the menu command hierarchy. *See* Tr. Ex. 1014. By April 1983, Stephenson installed at his client's business an operational version of his spreadsheet program, which had substantially the same menu hierarchy as the January 1983 version. That hierarchy is differently expressed from the hierarchy of both VisiCalc and 1-2-3. *See* Tr. Ex. 1019; Stephenson Affdvt. ¶¶ 56-60.

Stephenson first saw 1-2-3 in operation in February 1983, after he had developed the menu hierarchy for FIPS. Throughout the rest of that year, he continued to improve FIPS, changed its name VP-Planner, and began to consider marketing his program. By December 1983, Stephenson entered into a letter of intent with Adam Osborne regarding publication of VP-Planner. Osborne thereafter organized defendant Paperback Software. Stephenson Affdvt. ¶¶ 66-82.

Throughout 1984, defendants continued to improve VP-Planner. In the autumn, they recognized the success of 1-2-3 and reached the conclusion that spawned this litigation: VP-Planner, in order to be a commercial success, would have to be "compatible" with 1-2-3. "The only way to accomplish this result," defendants believed, "was to ensure that *the arrangement and names of commands and menus in VP-Planner conformed to that of Lotus 1-2-3.*" *Id.* at ¶ 117(emphasis added). *See generally id.* at ¶¶ 99-130. Such compatibility would allow users to transfer spreadsheets created in 1-2-3 to VP-Planner without loss of functionality for any macros in the spreadsheet. Also, such compatibility would allow users to switch from 1-2-3 to VP-Planner without requiring retraining in the operation of VP-Planner.

To some degree at least, defendants' premises have proved incorrect in hindsight. That is, first, as Excel has proved, a spreadsheet program did not have to be exactly compatible with 1-2-3 in order to be a commercial success. Second, copying the menu structure was not the *only* way to achieve aspects of this desired compatibility. For example, defendants could have instead added a macro conversion capability as the creators of Excel have successfully done (the Microsoft Excel Macro Translation Assistant), and could have provided an on-line help function that would show users the VP-Planner equivalent for 1-2-3 commands. *See Excel Reference* Guide at 491, 425-26 (Tr. Ex. 79). *See also* Morgan Decl. ¶¶ 3-7 (Docket No. 308) (Lotus itself created a "macro conversion utility" to translate macros among different-language editions of 1-2-3 (*e.g.*, North American, international English, French, German, Italian, Spanish. and Swedish)); Turner Decl. ¶ 10 (Docket No. 309). These points do not weigh significantly in the present decision, however, because even if VP-Planner olherwise would have been a commercial failure, and even if no other technological ways of achieving macro and menu compatibility existed, the desire to achieve "compatibility" or "standardization" cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual "work."

Defendants admit that, once these fateful decisions were made by Stephenson and Osborne, defendants converted VP-Planner into a program more like 1-2-3—indeed, a program that they have publicly advertised as a "workalike for 1-2-3." *VP Planner* Manual 1.11 (1985) (Tr. Ex. 9). It is incontrovertible that, in the process, they copied the expressive elements of 1-2-3 that the court has concluded are copyrightable:

> [M]aking the changes required for macro compatibility meant that we had to revise existing elements of the [VP-Planner] spreadsheet interface, including the hierarchical menu structure; ensure that keystroke sequences would bring about the same operational result in both programs; add certain functional elements found in Lotus 1-2-3 which VP-Planner did not yet support; and discard certain features which, although beneficial, were inconsistent with the macro compatibility requirement . . . .
>
> Several types of changes were required in the VP-Planner program to achieve keystroke macro compatibility. First, the menu structure had to be altered so that all menu commands would have the same first letter and be in the same location in the menu hierarchy as in Lotus 1-2-3.

Stephenson Affdvt. ¶¶ 144, 146. *See generally id.* at ¶¶ 142-157. After these changes were made, the VP-Planner manual could truthfully declare:

> VP-Planner is designed to work like Lotus 1-2-3, keystroke for keystroke. . . . VP-Planner's worksheet is a feature-for-feature workalike for 1-2-3. It does macros. It has the same command tree. It allows the same kind of calculations, the same kind of numerical information. Everything 1-2-3 does, VP-Planner does.

VP-Planner Manual at xi. 1.11.

The court's comparison of the 1-2-3 menu command hierarchy and the VP-Planner menu hierarchy confirms that VP-Planner "has the same command tree" as 1-2-3—that is, that defendants copied the expression embodied in the 1-2-3 menu

hierarchy. It is true that there are some differences between 1-2-3's menu structure and VP-Planner's menu structure. For example, most VP-Planner menu lines begin with a help ("?") command, and some additional commands are included at the end of some menu lines (*i.e.*, "DBase, Multidimensional" on the "/File Erase" menu line; and "Page #, No Page #, Row/Col. #, Stop Row/Col. #, Background" on the "/Print Printer Options Other" menu line). Other differences between the two programs appear in the start-up screens, the placement on the screen of the menu lines, the exact wording of the long prompts, the organization of the help screens, the increased width of the VP-Planner screen, and the ability of VP-Planner to hide certain columns. *See* Stephenson Affdvt. ¶¶ 161-176. The works are, nevertheless, substantially, indeed, strikingly, similar. As Judge Learned Hand held in a copyright case involving a pattern on a bolt of cloth that was used to make dresses, infringement may be found despite some differences between two works:

> the ordinary observer, unless he set out to detect the disparities, would be disposed to overlook them, and regard their aesthetic appeal as the same. That is enough; and indeed, it is all that can be said, unless protection against infringement is to be denied because of variants irrelevant to the purpose for which the design is intended.

Peter Pan Fabrics, 274 F.2d at 489, *quoted in part with approval, Concrete Machinery*, 843 F.2d at 607. *See also Atari v. Philips*, 672 F.2d at 618 ("[A] laundry list of specific differences . . . will not preclude a finding of infringement where the works are substantially similar in other respects. . . . When analyzing two works to determine whether they are substantially similar, courts should be careful not to lose sight of the forest for the trees.").

From the perspective of both an expert and an ordinary viewer, the similarities overwhelm differences. Thus, as in *Peter Pan Fabrics*, the two works at issue are substantially similar. Indeed, by using the option in VP-Planner that allows a user to move, the menu from the bottom of the screen to the

top of the screen' a user could easily think 1-2-3 rather than VP-Planner was the program in use. Certainly purchasers of a book designed to teach users how to master 1-2-3, which is distributed with demonstration copies of *VP-Planner*, would be likely to overlook the disparities between 1-2-3 and VP-Planner. *See* L. Ingalsbe, *Lotus 1-2-3 with Version 2.0 for the IBM PC* (1987) (Tr. Ex. 215).

Moreover, even if some elements of VP-Planner were very different, it would not give defendants a license to copy other substantial elements of 1-2-3 verbatim. If one publishes a 1,000-page book of which only a 10-page segment is an unauthorized reproduction of copyrighted material, and if the 10-page segment is a qualitatively substantial part of the copyrighted work, it is not a defense to a claim of infringement that the book is 99% different from the copyrighted material. *Sheldon*, 81 F.2d at 56 ("no plagiarist can excuse the wrong by showing how much of his work he did not pirate"); *SAS Institute*, 605 F.Supp. at 829-30. Thus, defendants' proof that VP-Planner has many features that are different from Lotus 1-2-3 is off point. The more relevant question is: does it have significant features that are substantially similar? I conclude, on the record before me, that there is no genuine dispute of material fact on this question. The answer to this question must be "yes."

Accordingly, I conclude that it is indisputable that defendants have copied substantial copyrightable elements of plaintiff's copyrighted work. I therefore conclude that, subject to consideration of other contentions advanced by defendants, liability has been established.

VII. A POSTSCRIPT ON THE NATURE OF DECISIONMAKING IN THIS CASE

A. *Policy Arguments and Limitations on the Role of Courts*

Although Parts I through VI of this Opinion are sufficient to demonstrate that defendants have impermissibly copied copyrightable elements of expression embodied in 1-2-3, it may be useful to pause, before reaching the merits of defendants' proffered defenses to liability, to consider the proper use of policy in determining statutory meaning.

As noted in Part III(A)(1), *supra*, the court is directed to look to the "object and policy" of the law in determining the manifested meaning of the law. *Kelly*, 479 U.S. at 43, 107 S.Ct. at 358. The court must, however, be careful that its weighing of the "object and policy" implicit in the law does not override the mandates that are either explicit in the statutory text or implicit in the provisions of the whole law.

Defendants' arguments in this case have urged the court to violate that limitation on the judicial use of policy considerations. That is, in the name of the "object and policy" of copyright law, defendants have argued: (1) that the idea-expression distinction creates unnecessary and counter-productive uncertainty that could be avoided by a bright-line test of copyrightability; (2) that the policy of promoting creativity favors extremely narrow copyright protection in computer programs for anything beyond source code and object code; and (3) that the policy of encouraging standardization mandates the conclusion that a user interface is not copyrightable. Also, in advancing these policy arguments, defendants have obscured the nature of important issues in the case by strained analogies and word games.

Policy arguments advanced without focusing on whether they are inconsistent with the mandates of legislation—along with strained analogies and word games—are fundamentally flawed because they do not take account of limitations on the role of courts, to which counsel as well as courts are obliged to be sensitive. Courts must enforce the mandates of legislation, and even when answering questions of law not answered by those mandates, must do so in a way that respects legislative policy choices, whether explicit or implicit.

Each of defendants' policy arguments addressed below suffers from this fatal flaw.

B. *Strained Analogies and Word Games*

The reasoning that lawyers propose in arguments and judges use in deciding cases often builds on simile, metaphor,

and allusion. All such reasoning is within the scope of what we call analogy, in the broadest sense. When strictly logical deductions from authoritative declarations of law—in constitutions, statutes, and precedents—do not provide answers to questions of law that must be decided in the case before the court, the court turns to analogy. *Cf. Kelly*, 479 U.S. at 43, 107 S.Ct. at 358 ("we must . . . look to the provisions of the whole law"). Thus, similes, metaphors, and other forms of allusion are appropriately a part of our efforts to communicate ideas. They are not to be mistaken, however, for logically compelled inferences from authoritative declarations. They are aids to understanding the meaning, not yet authoritatively determined, of the authoritative declarations as applied to circumstances those declarations did not explicitly address.

Defendants strongly urged the court to accept an analogy between a computer program and a "useful article." As explained in Part IV(A) of this Opinion, that proposed analogy is deeply flawed because of an unstated implicit premise that the fact that a computer program is "useful" utterly precludes copyrightability of any expressive elements in it. In the same breath, though, defendants concede, that source code and object code are copyrightable. This concession is itself a demonstration of the vulnerability of the "useful article" argument. The concession makes clear a flaw in the analogical argument for treating a computer program that is "useful" as necessarily nothing more than a "useful article." If no explanation is offered for the exception other than precedent, the analogy itself is challenged by the precedent.

Thus, labeling a computer program as a "useful article" in relation to a claim for copyrightability of nonliteral elements of the user interface, but not a "useful article" in relation to a claim for copyrightability of source code and object code, is engaging in a word game rather than focusing on the merits of the copyrightability issue.

An even more striking word-game argument is defendants contention that in copying the 1-2-3 user interface, they have only copied a "language," and that languages are not copyrightable. *See, e.g.*, Deposition of Steven Cook at 153:23-

154:10 (Docket No. 327); Kocher Affdvt. ¶¶ 13-18 (Docket No. 272); Defendants' Post-Trial Brief at 87-103 (Docket No. 325). Although defendants have not explicitly stated all the essential steps of this argument, by filling the gaps in trying to understand the argument, one may infer the following elements and corollaries of the argument:

(1) Although expression is copyrightable, the language in which the expression is written is not copyrightable. Thus, a book written in English or French may be copyrightable, but the English and French "languages" are not works in which copyright may subsist.

(2) Like books, computer programs, written in computer programming "languages," may be copyrightable, but only the "sets of statements or instructions," and not the "language" in which they are written, are copyrightable.

(3) 1-2-3 has a macro capability or facility that allows a user to "write" her or his own macros. When the user writes a macro, he or she is actually writing a kind of computer "program." *See* 1-2-3 Release 1A User's Manual at 117 (macro "facility gives the 1-2-3 user a true programming capability") (Tr. Ex. 72); 1-2-3 Release 2.01 User's Manual at 3-10 ("1-2-3 includes a number of macro commands that create a powerful programming language") (Tr. Ex. 72B).

(5) The macro, or "program," that the user writes may be copyrightable if original and nonobvious. The "language" in which the macro is written is never copyrightable.

(6) Thus, when defendants copied the menu command hierarchy from 1-2-3, they did not copy a copyrightable element that embodied expression, but rather, copied only the "macro facility language" of 1-2-3, a non-copyrightable element in the public domain.

The vulnerable steps of this argument include, at a minimum:

(1) That "language" has a single, invariable meaning in all discourse about "languages";

(2) That not only languages such as English and French but all other languages as well—including Esperanto, and *Reiss'* coined words, 276 F. at 718, and Pascal—are automatically ineligible for copyright protection (as to vulnerability of this step of the argument, defendants, though invited at trial to do so, have cited no precedent that supports the contention that a "language," even if original, is not copyrightable);

(3) That "language" and "sets of statements or instructions" are opposites, and never the twain shall meet—that is, anything that is a "set of statements or instructions" is not a "programming language," and anything that is a "programming language" is not "a set of instructions."

Just to state these implicit elements of defendants' argument is sufficient to demonstrate that the argument depends on arbitrary definitions of words, adopted for undisclosed reasons. In human experience, words have variable meanings; dictionaries seldom list a single definition for a word. An argument that depends on the proponents' undisclosed definitions of words—and even different definitions as a word is used in different steps of the argument—becomes a word game that obscures the substantive meaning of the argument and is an obstruction rather than an aid to the court's use of the adversary process to inform and thus improve decisionmaking.

Having explored the argument fully to try to understand its true nature, I conclude that defendants' "language" argument about the macro facility of Lotus 1-2-3, like defendants' "useful article" argument examined in Part IV(a), *infra*, is totally without merit.

C. *Policy Arguments for Bright-Line Rules*

One of the themes in defendants' bundle of policy arguments is a plea for bright-line certainty. Indeed, it is a persistent theme throughout defendants' submissions: Software developers would like to know what they may and what they must not copy. The idea-expression distinction, defendants argue, fails to achieve this policy goal; therefore, the court

should adopt instead the bright-line literal-nonliteral distinction.

The yearning for certainty is not unique to this case or to copyright law. But certainty has a price. It reflects a hope that the court will apply hard-and-fast rules despite circumstances that cry out for a judgmental standard of decisionmaking that takes account of circumstances an impartial observer would think relevant to a "just" disposition of the case—which, after all, is one of the declared objectives of the legal system. *E.g.* Fed.R.Civ.P. 1 ("just, speedy, and inexpensive determination of every action").

The more deeply one probes, whether as lawyer or judge, or as an interested observer or critic of the legal system, the clearer it becomes that in many circumstances hard-and-fast rules, despite their initial attractiveness and false promise of certainty, have consequences that offend one's sense of justice. For example, hard-and-fast rules, because they allow for little or no evaluation and discretion, sometimes fail to take account of the competing values underlying the relevant law, and fail even to attempt to find an accommodation that serves conflicting high-value interests as well as possible, and at the lowest possible detriment to each. Professionals in law are forced by the nature of their work to do more probing into the disadvantages of bright-line rules than other observers generally are likely to do. For this reason, professionals are less likely than other observers to expect decisionmaking based on hard-and-fast rules that, in hindsight, seems arbitrarily to disregard relevant circumstances and to deserve the pejorative label "legalistic."

In this case, defendants have nevertheless proposed a bright-line rule that would extend copyright only to source code and object code, and never to nonliteral elements of computer programs. That proposal must be rejected. Not only is the proposed distinction contrary to the statutory mandates discussed throughout this Opinion, but also, it reflects the false hopes that are endemic to trying to apply brightline rules where judgmental, evaluative standards are more appropriate. *Nichols*, 45 F.2d at 121; *Peter Pan Fabrics*, 274 F.2d at 489.

In any event, as noted in Part II, Congress could have constitutionally drawn defendants' proposed bright-line—providing protection only to computer program code and not to nonliteral elements of computer programs—but Congress has not done so. Without a congressional mandate, it would be an abuse of authority for this court, in deciding this case, to use a bright-line test of copyrightability that makes the literal-nonliteral distinction decisive. Instead, the court must adjudicate within the area of protection mandated by Congress.

D. *Opinion Evidence and Premises of Legal Rulings*

Defendants, though disavowing any intent to invite the court to engage in lawmaking, have gone to extraordinary lengths in this case to advance policy arguments that a court can appropriately consider only if, in order to decide the case *sub judice*, it must answer previously unanswered questions about the meaning of relevant constitutional, statutory, and judicial sources of authority. Whether such a determination of previously undetermined meaning of an authoritative declaration is described as lawmaking, as I believe it to be, or instead in some other way—for example, as determining what "old statutes have since become" in circumstances materially different from those existing when the statute was enacted, R. Dworkin, *Law's Empire*, 248-50 (1986)—is of less consequence than how a court goes about deciding meaning for the case at hand. What "facts" or "evidence" shall the court consider? Shall the court consider expert opinions about what the circumstances were at the time the statute was enacted, and how they differ now?

Defendants' submissions have failed to address directly these basic questions about the appropriate role of courts. Implicitly, however, they have asked the court to consider policy arguments as freely as if the court were under no constraints in this respect. Defendants, for example, offered the following opinion testimony of a witness qualified as an expert in computer software development:

> If the law is interpreted to say that a developer may not make a new program that can use the data files and

> inputs accepted by existing programs, then computer advancement as we know it will be slowed. . . .
>
> I believe that such a decision would throw the entire software industry into confusion. . . . [T]here would be a chilling effect on development and advancement in many areas. . . .
>
> This case has implications about whether or not new, improved programs can read and execute data users created for themselves with an older program. The quantity of data to be affected by this decision is incomprehensively vast. It is not a case about "cheap copies".
>
> If aspects of screen displays that are governed by functionality (such as a command language like the 1-2-3 command structure) are held to be within the scope of copyright protection, then progress in application and systems computer programs could be dramatically slowed in the United States.

Bricklin Affdvt. ¶¶ 176-77, 179-80. *See also* Samuelson & Glushko, *Comparing the Views of Lawyers and User Interface Designers*, 30 Jurimetrics at 137 (concluding on basis of survey that leading user interface designers oppose strong copyright protection for elements of a user interface because such protection would be harmful, rather than helpful, to the industry).

By rulings during trial, the court received opinion evidence of this kind into the "record" in the broadest sense, even though excluding it from consideration in relation to any material adjudicative fact that might be in dispute. Not surprisingly, defendants, though explicitly invited to do so, were never able to identify any "adjudicative fact" question, as that term is used in Rule 201 of the Federal Rules of Evidence, to which opinion evidence such as this would be relevant. The court nevertheless received this evidence, over plaintiff's objection on the ground of irrelevance, because courts are free to consider "evidence" of this type when deciding "legislative" (or "premise") facts. That is, courts may consider

such evidence to resolve disputable assertions of an evaluative nature (either about the past or about the future) or assertions about historical facts, or predictions about future consequences of adopting one or another rule of law, as premises for deciding what legal ruling to make. *Cf.*: F.R.E. 201(a), Notes of Advisory Committee (explaining "legislative facts"). *See also In re Asbestos Litigation*, 829 F.2d 1233, 1245-52 (3d Cir.1987) (concurnng opinion of Becker, J.), *cert. denied*, 485 U.S. 1029, 108 S.Ct. 1586, 99 L.Ed.2d 901 (1988). This is not to say, however, that either the trial court that allows the evidence to be received into the record or an appellate court on review is limited by the rules of admissibility and the standards of review that apply to disputed issues of "adjudicative" fact. *Chastleton Corp. v. Sinclair*, 264 U.S. 543, 548, 44 S.Ct. 405, 406, 68 L.Ed. 841 (1924) ("the Court may ascertain as it sees fit any fact that is merely a ground for laying down a rule of law"). *Cf. Lockhart v. McCree*, 476 U.S. 162, 168-69 n. 3, 106 S.Ct. 1758, 1762-63 n. 3, 90 L.Ed.2d 137 (1986) ("We are far from persuaded, however, that the 'clearly erroneous' standard of Rule 52(a) applies to the kind of 'legislative' facts at issue here."). *See also Dunagin v. City of Oxford, Mississippi*, 718 F.2d 738, 748 n. 8 (5th Cir.1983) (en banc) ("legislative" fact discussion by Reavley, J.) ("If the legislative decision is not binding at this stage, at least it carries great weight. Certainly it cannot be thrust aside by two experts [who were called as witnesses before the trial court] and a judicial trier of fact."), *cert. denied*, 467 U.S. 1259, 104 S.Ct. 3554, 82 L.Ed.2d 855 (1984).

The court having received defendants' proffered evidence into the record for the limited purpose of considering its weight, if any, as bearing on premises of any legal ruling the court might find it necessary to make in deciding the case, plaintiff responded with sharply contrasting opinions in its post-trial brief:

> *First*, the tremendous growth and success of the U.S. software industry is the direct result of the creative and original efforts of its software developers, laboring under

> the protection of the copyright laws. Innovation has been the key to market success. . . .
>
> *Second*, to the extent that defendants attempt to characterize this struggle as pitting large heartless corporate giants against lonely and defenseless developers working out of their dens and basements, the defendants do not understand what is at stake here. If the elements of a computer program at issue here were to lose their copyright protection, the biggest losers of all would be the small developers. The history of this industry has been one of creative designers who identify an unfilled need in the market and then design and build a superior product to fill that need. . . . [T]he developers' ability to realize substantial rewards for their creative efforts has depended entirely upon the legal protection copyright has afforded their work.
>
> If the defendants could rewrite the law, however, those days would be over. The first time a developer demonstrated an attractive new product at Comdex or Softeach or some other industry gathering, hundreds of programmers in corporate research laboratories around the world would set to work creating their own versions of the program to compete with the original. A major firm, with a staff of talented programmers, could fairly promptly create a clone of almost any new program and, in so doing, rob the author of much of the value of his creative efforts. The original author might have a head start, but that would provide little comfort once the major firms hit the market with their "new" products. . . . If copying were legal, the creators would lose out. It is as simple as that.
>
> It is no accident that the world's strongest software industry is found in the United States, rather than in some other jurisdiction which provides weaker protection for computer programs. The system is working, and there is no reason to change it.

Plaintiff's Post-Trial Brief at 87-89; *see also* Tarter Decl. ¶¶ 72-73 (Docket No. 289) ("[I]nterface design represents one of the most crucial aspects of software design. In order to promote innovation and the evaluation of better software products, it is important that we protect the investment that the best developers make in interface design."); Clapes, *Software, Copyright, & Competition* at 202-03 (arguing that the user interface, which is often the most important feature of a computer program. requires strong copyright protection to encourage further innovation and advancement).

The opinion testimony offered in this case failed completely to focus on circumstances existing at any time Congress acted to enact or amend any part of the copyright law. For this reason, it has little, if any, value in illuminating issues of law presented in this case. It is relevant, of course, that both the text and history of the copyright law manifest a purpose of encouraging innovation and of doing so through copyright protection. At the time of congressional action. however—even as late as the 1976 Act and the 1980 amendments—it was not possible to know how the computer programming industry would change over the ensuing years, and what exact scope of protection would best advance the policy goals in newly developed and now foreseeably developing circumstances. In light of that uncertainty, shall the court consider expert opinions about what the law ought to be today? What if those opinions appear to be inconsistent with the accommodation, struck by the statute now in effect, among competing and to some extent conflicting public policy justifications for broader or narrower copyright protection?

I conclude that a court would be going well beyond even the precedents most receptive to consideration of legislative history if it took into account expert opinions of persons experienced in the field of computer software development that are manifestly inconsistent with the accommodation that Congress struck when it enacted the Copyright Act of 1976 and the most recent amendments of the copyright law. In the 1976 Act, Congress manifested an intention to encourage innovation in the computer programming field, and to do so

through copyright law that explicitly gives substantial protection to innovative *expression* in intellectual "works," including computer programs. Congress made this decision despite expert testimony that extending copyright protection to computer programs might have "disastrous consequences . . . on standardization in electronic data processing," *Copyright Law Revision: Hearings on S. 597 Before the Subcommittee on Patents, Trademarks, and Copyrights of the Senate Committee on the Judiciary*, 90th Congress, 1st Session 589 (1967) (Testimony of Professor Anthony Oettinger), despite academic criticism of the decision to extend copyright protection to computer programs, Breyer, *Uneasy Case for Copyright*, 84 Harv.L.Rev. at 340-351 (concluding that "[c]omputer programs should not receive copyright protection at the present time"), and despite explicit proposals that copyright protect only "the instructions themselves" while leaving others free to otherwise "replicate [another's] program exactly." *Hearings on S. 597* at 571-73 (Statement of EDUCOM) ("Broad copyright protection for programs is unwise and improper . . . [H]ad programming been constantly carried out under the threat of infringement actions charging plagiarism of existing copyrighted programs, it is doubtful whether the grouth of programs and programming techniques of recent years would have been possible."). The 1980 amendments, too, plainly went farther in protecting the interests of the creators of an original software product than defendants' experts now propose (and than some experts at the time proposed, *e.g.*, CONTU, *Final Report* at 56-76 (Hersey, C., dissenting)), even though they did not go as far as proposed by others to whose views Congress then had access.

When a court determines the meaning of the copyright statute in a new context, different from what could have been known to even the most prescient observer at the time Congress acted, the court must respect the congressional mandate—especially with respect to congressional determinations of premise facts—and is constrained to search not for what the court might now independently consider to be the best accommodation of competing interests but instead to deter-

mine an accommodation in this new context that is consistent with the accommodation that Congress enacted for the context to which it spoke. To the extent that expert opinions illumine the comparison between the context to which Congress spoke (the circumstances of software development then existing and predictable future circumstances) and the current context (the changing circumstances of software development now existing and now predictable for the future) the court may and should take them into account. To the extent, however, that expert opinions propose solutions inconsistent with the congressional accommodation, they must be disregarded.

On the basis of the complete record now before me, I conclude that defendants' policy arguments in this case are inconsistent with the accommodation struck by Congress in passing the Copyright Act and its amendments. Moreover, defendants have not demonstrated any significant changes in the premise facts upon which Congress acted, and upon which CONTU rested its recommendations. If anything, user interfaces have become more expressive and more communicative since 1976 and 1980. I thus conclude that I must disregard defendants' experts' prediction of doom for the computer programming industry if copyright is extended to the user interface and other nonliteral elements of computer programs that embody expression, just as Congress disregarded Professor Oettinger's gloomy prediction of "disastrous consequences" and Commissioner Hersey's dissent. Rather, this legal issue must be resolved in such a way as to extend copyright protection, clearly and unequivocally, to those nonliteral elements of computer programs that embody original expression.

E. *Defendants' Policy Arguments Founded On the OTSOG Principle*

One more of defendants' policy arguments deserves attention. Despite statutory mandates supporting the conclusion that elements of the user interface of 1-2-3 are copyrightable, defendants argue that the need to achieve compatibility and standardization compels rejection of that conclusion on pol-

icy grounds. Copyrightability of a user interface, they argue, will frustrate the public interest in allowing programmers to achieve innovation by "borrowing" and improving upon ideas of other programmers, and will undermine attempts to achieve compatibility and standardization among different programs. Especially in the vital area of user interfaces, defendants contend, copyrightability will have adverse consequences on encouraging innovation and on the broader public welfare.

Defendants' general contention—that "Progress of Science and useful Arts" cannot occur unless authors and inventors are privileged to build upon earlier progress and earlier innovation—has long been a virtually unchallenged premise in all branches of the law of intellectual property. An early expression of the point is Newton's declaration: "If I have seen further it is by standing on ye sholders of Giants." Sir Isaac Newton, Letter to Robert Hooke, February 5, 1675/1676, *quoted in* R. Merton, *On the Shoulders of Giants: A Shandean Postscript* 31 (1965).[3] This principle, which Merton has referred to as "OTSOG" (based on the modernized "on the shoulders of giants"), *id.* at 270, is also firmly established in our case law:

> In truth, in literature, in science and in art, there are, and can be, few if any, things, which in an abstract sense, are strictly new and original throughout. Every book in literature, science and art, borrows and must necessarily

[3] Newton's now-famous phrase—"on ye shoulders of Giants"—may have been copied, at least in its "total concept and feel," from Bernard of Chartres who, according to Merton, authored the OTSOG aphorism in the early twelfth century when he taught: "we are like dwarfs standing [or sitting, for those dwarfs who prefer safety to far-ranging vision] upon the shoulders of giants, and so able to see more and see farther than the ancients." Merton, *OTSOG* at 178-92. As Merton demonstrates, however, Bernard himself was standing on the shoulders of Priscian, a sixth-century grammarian, who wrote: "The younger [*e.g.*, the more recent] the scholars, the more sharp-sighted." *Id.* at 194-95 (brackets added by Merton). Of course, Prisclan's passage presents only the idea; it took Bernard to express that idea in the metaphor of giants and dwarfs.

borrow, and use much which was well known and used before.

Emerson v. Davies, 8 F.Cas. 615, 619 (C.C. D.Mass.1845) (No. 4,436) (Story, J.). *See also Ecclesiastes* 1:9-10 ("[T]here is no new thing under the sun. Men may say of something, 'Ah, this is new!'—but it existed long ago before our time."). Defendants, however, attempt to squeeze something from the OTSOG cornerstone that is not there.

Two possible applications of OTSOG in the field of computer programming are relevant to this case. First, innovation in computer programming is advanced as each programmer builds upon the ideas of previous programmers. Second, some of the innovative ideas may be expressed in a particular way that is so effective or efficient that the expression becomes standardized throughout the field even though the idea is capable of being expressed in other ways—that is, even though the particular expression is not an essential detail to every expression of the idea. Although both of these corollaries of OTSOG are important to the future of computer programming, neither was embraced by Congress (as explained below) in such a way as to override the public interest in conferring upon an author a right to a limited monopoly in the author's "work."

The metaphorical "shoulders of giants" on which successors may legally stand are not as broad as defendants contend. The legally relevant shoulders of programming giants are their ideas—and do not extend to all of their expressions. The encouragement of innovation requires no more. It is sufficient that programmers are privileged to borrow and improve upon previous ideas—such as the ideas for an electronic spreadsheet and a two-line moving cursor menu. Adequate room for innovation remains even though successors are barred from copying earlier authors' particular expressions—such as the particular structure, sequence, and organization of a menu command system. *Pearl Systems*, 8 U.S.P. Q.2d at 1525.

Of course, if a previous programmer's idea can be expressed in only one or a limited number of ways—such as the rotated "L" screen display or the use of the slash key to

invoke the menu command system—then the expression, too, may be copied. *Morrissey*, 379 F.2d at 678-79. Where, in contrast, the idea is capable of countless ways of being expressed, only inexpensive cloning, and not innovation, would be advanced by allowing programmers to copy the particular way the ideas have been expressed by others.

The second corollary of the "OTSOG" principle relevant to this case concerns standardization. Defendants have argued that 1-2-3, and specifically, 1-2-3's menu structure and macro command facility, has set a *de facto* industry standard for all electronic spreadsheets. Thus, defendants had no choice, they argue, but to copy these expressive elements from 1-2-3. Had they not copied these elements (including the macro facility), users, who had been trained in 1-2-3 and had written elaborate macros to run on 1-2-3 spreadsheets, would be unwilling to switch to VP-Planner. VP-Planner would be a commercial failure. Neither the factual nor the legal predicate of the argument is supportable.

First, defendants' argument ignores the commercial success of Excel, an innovative spreadsheet program that is not compatible with 1-2-3, either in its menu structure or in its macro command facility. Also, defendants argument ignores the alternatives to direct copying that were legally available to them.

As already explained, to the extent that VP-Planner was concerned with compatibility for macros written originally for 1-2-3, VP-Planner could have provided for a translation device that could read 1-2-3 macros and convert them, automatically, into macros that could be run on VP-Planner. Microsoft Corporation successfully included such a capability in Excel and Lotus itself has written such a capability for translating macros among different-language versions of 1-2-3. Defendants have not offered persuasive evidence to show that they could not have done the same with VP-Planner. That "[i]t would have been an extremely complicated task" and would have cost defendants more to do so, Dauphinais Affdvt. ¶ 163, is not a reason for denial of copyright protection to 1-2-3. Copyright protection always has consequences of this kind.

Moreover, to the extent that VP-Planner is an incremental improvement upon 1-2-3 because of its multidimensional database capabilities, defendants could have (1) sought a license to use 1-2-3's menu structure and command facility; (2) offered to sell their new expression of ideas to Lotus for inclusion in future versions of 1-2-3; or (3) marketed VP-Planner as an "add-in" to 1-2-3.

An "add-in" program is a program designed to be used in conjunction with another program. For example, HAL is a program that is used in conjunction with 1-2-3—a user must have both HAL and 1-2-3—which allows a 1-2-3 user to enter commands in simple sentences such as "copy column B to E" rather than by selecting commands from the menus. Tr. Ex. 176. To encourage the creation of add-in programs (which in turn, make 1-2-3 a more attractive program to potential customers), Lotus has published a product called Lotus Developer Tools that helps developers write add-in programs. Tr. Ex. 138. Thus, even if defendants found the first two alternatives unattractive or unattainable (for example, because of Lotus' nonconsent except on unfavorable terms, *see Stewart*, 110 S.Ct. at 1764 ("a copyright owner has the capacity arbitrarily to refuse to license one who seeks to exploit the work")), they could have marketed their multidimensional database capability as an "add-in" for those users interested in that capability. Transaction and marketing costs of the third option would tend to screen out the new product unless its innovative features made it very attractive to potential users, but this functional effect of copyright law would be entirely consistent with the objects and policies of that law as manifested in the copyright statute and judicial precedents. By instead selling a stand-alone product that completely replaces 1-2-3, defendants have not merely sold and profited from only their incremental additional expression. Rather, they seek permission to profit also from copying Lotus' protected expression.

Defendants' standardization argument is flawed for another reason as well. As explained above, one object of copyright law is to protect expression in order to encourage innovation.

It follows, then, that the more innovative the expression of an idea is, the more important is copyright protection for that expression. By arguing that 1-2-3 was so innovative that it occupied the field and set a *de facto* industry standard, and that, therefore, defendants were free to copy plaintiff's expression, defendants have flipped copyright on its head. Copyright protection would be perverse if it only protected mundane increments while leaving unprotected as part of the public domain those advancements that are more strikingly innovative.

Finally, the entire argument about standardization may be questioned on a more fundamental ground. Defendants have cited no statutory provision or precedent that has ever declared that standardization, when not achieved *de jure*, is necessarily in the public's best interest. The court is aware of no such precedent or legislative mandate. A moment's reflection is enough to disclose that the public interest in extensive standardization is a sharply debatable issue. *See, e.g.*, Tarter Decl. ¶¶ 34-45; *cf.* the QWERTY typewriter keyboard (discussed in Farrell, *Standardization and Intellectual Property*, 30 Jurimetrics 35, 36-38 (1989)). Decisive against defendants' contention, in any event, is that the particular way they propose that the court resolve this dispute would reduce copyright protection far below the mandate of the copyright act.

VIII. OTHER DEFENSES

A. *Lack of Subject Matter Jurisdiction*

Section 411 of Title 17 provides that an infringement action may not be brought unless the author has applied for registration of the copyright. If the application is denied, the infringement action may not be brought unless the author serves a copy of the complaint on the Register of Copyrights. "Copyright registration under § 411(a) is a [jurisdictional] condition precedent to filing an infringement action." *Quincy Cablesystems, Inc. v. Sully's Bar, Inc.*, 650 F.Supp. 838, 850 (D.Mass.1986).

Defendants argue that plaintiff's allegations of copyright infringement in 1-2-3's user interface are, in reality, allegations of copyright infringement in 1-2-3's "screen displays." Defendants further argue that plaintiff's registration certificate for 1-2-3 covers only the literal code of 1-2-3, that plaintiff has failed to register the "screen displays" of 1-2-3 as separate audiovisual works, and that this court consequently lacks jurisdiction over allegations of infringement of the "screen displays" of 1-2-3.

This contention borders on the frivolous. First, I emphatically reject defendants' premise, based on yet another word game, that equates the user interface of 1-2-3 with 1-2-3's "screen displays." In ruling that defendants have infringed plaintiff's copyright, I have not ruled that defendants are liable because they copied the "screen displays" of 1-2-3. Rather, I have concluded that defendants copied protected nonliteral elements of expression in the user interface and the underlying computer program.

Second, I conclude that, in any event, plaintiff's registration certificates are sufficient to extend to the "screen displays" of 1-2-3.

By focusing only on "screen displays," defendants have failed to take due notice of the statutory language as it bears on whether copyrightable elements of computer programs include the "user interface" and other nonliteral aspects.

Depending on the sense in which the phrase "screen display" is used, it may be quite correct to say that

> copyright protection of a computer program does not extend to screen displays generated by the program. . . . This distinction results from the fact that the same screen can be created by a variety of separate and independent computer programs. It is somewhat illogical to conclude that a screen can be a "copy" of many different programs.

Softklone, 659 F.Supp. at 455-56.

It is not, though, the screen display itself, in this narrow sense, that is a copyrightable "computer program"—that is, "a

set of statements or instructions . . . to bring about a certain result," 17 U.S.C. § 101. The screen display, understood in this narrow sense, is useless for this purpose unless it is integrated with other parts of the computer program so as to become indirectly part of the means by which the user communicates instructions through the total program to bring about the desired result. Thus, the copyrightable element is not, strictly speaking, the screen display, narrowly understood, but the literal and nonliteral elements of not only the display but also the distinctive way of creating it. Indeed, the *Softklone* court recognized this point:

> Therefore, it is this court's opinion that a computer program's copyright protection does not extend to the program's screen displays, and that copying of a program's screen displays, without evidence of copying of the program's source code, object code, *sequence, organization, or structure*, does not state a claim of infringement.

659 F.Supp. at 456 (emphasis added).

Accordingly, just as using different dialogue does not sanitize from claims of infringement a play that copies the plot and characters from some other copyrighted work with resulting substantial similarity, the fact that the screen display *could* easily be created in many other ways—either to perform the same or to perform very different functions—does not sanitize it from claims of copyright infringement. The critical question, instead, is whether copyright protection is available against creating and using such a screen display—including the manifestation of structure, sequence, and organization in that display—as a part of the "set of statements or instructions" in a program that is designed "to be used directly or indirectly in a computer in order to bring about a certain result."

In any event, it is appropriate to consider "screen displays" in a broader sense that also includes the structure, sequence, and organization of the underlying program as manifested in the menus presented on the screen displays. I conclude that plaintiff's certificates of copyright registration in the "entire

work" of 1-2-3 are sufficient to extend copyright protection to the "screen displays," in this broader sense, of Lotus 1-2-3. *See* TX 1-233-501 (Tr. Ex. 1, covering "entire work" of "1-2-3"), TX 1-233-502 (Tr. Ex. 2, covering "entire work" of "1-2-3 Release 1A"), and TX 1-694-602 (Tr. Ex. 3, covering "computer program text" of "1-2-3 Release 2").

The Copyright Office registers different works in different ways. One registration form may be used for nondramatic literary works (Form TX), different forms are required for audiovisual works (Forms PA and VA), and a different form for sound recordings (Form SR). *Compendium of Copyright Office Practices* § 604 (1984) (hereinafter "*Compendium II*"). *See also* 37 C.F.R. § 202.3(b)(1) (1989). As Congress recognized, however, the seven distinct categories of copyrightable works that are listed as examples in section 102(a) are not mutually exclusive:

> The items are . . . overlapping in the sense that a work falling within one class may encompass works coming within some or all of the other categories.

House Report at 53, *repnnted at* 5666. Thus, in registering a work that defies easy categorization, an author is directed to use the application form "in the class most appropriate to the type of authorship that *predominates* in the work being registered." *Compendium II* § 604 (emphasis added); 37 C.F.R. § 202.3(b)(2). *See also Compendium II* at § 708 ("The appropriate application form is generally determined by the nature of the authorship in which copyright is claimed."); 37 C.F.R. § 202.3(b)(2).

Most computer programs, like 1-2-3, are registered on Form TX as nondramatic literary works. *See Compendium II* at § 702.01 ("computer programs" should be registered as "nondramatic literary works" on Form TX). Some computer programmers, however—notably authors of video games—have instead registered for copyright the screen displays created by the program. *Stern Electronics*, 669 F.2d at 855 (plaintiff "eschewed registration of its program as a literary work and chose instead to register the sights and sounds of

[the video game] as an audiovisual work"); *M. Kramer Manufacturing Co. v. Andrews*, 783 F.2d 421, 442 (4th Cir.1986) ("a copyright in the audiovisual display, which display is created by a computer program, protects not only the audiovisual from copying, but also the underlying computer program to the extent the program embodies the game's expression"). *See also, Atari Games v. Oman*, 888 F.2d at 882 ("Video games, case law confirms, rank as 'audiovisual works' that may qualify for copyright protection."); *Midway Manufacturing Co. v. Artic International, Inc.*, 704 F.2d 1009, 1012 (7th Cir.) (same), *cert. denied*, 464 U.S. 823, 104 S.Ct. 90, 78 L.Ed.2d 98 (1983). This form of registration is particularly appropriate where the chief function, and the bulk of creative expression, of a particular computer program is the creation of a series of pictorial images on the computer screen as with video games.

Other authors of computer programs have applied for dual registrations for computer programs—one "literary" registration to cover the code, and a separate "audiovisual" registration to cover the screen displays. *See Softklone*, 659 F.Supp. at 455-56. This practice has since been rejected by the Copyright Office as duplicative. *See* Library of Congress, Copyright Office, *Registration of Computer Screen Displays*, Docket No. 87-4, 53 Fed.Reg. 21,817 (1988) ("all copyrightable expression owned by the same claimant and embodied in a computer program, including computer screen displays, is considered a single work and should by registered on a single application form"); Copyright Office, "Computer Programs and Related Screen Displays" (Tr. Ex. 1179) (single registration required for computer programs and their screen displays, including video game computer programs); *Compendium II* at § 607 (single registration covers single unitary work); *id.* at § 609 (copyright owner may "make only one basic registration per work"); 37 C.F.R. § 202.3(b)(6) (1989) (same); *Manufacturers Technologies*, 706 F.Supp. at 993; *Broderbund*, 648 F.Supp. at 1133.

In any event, when Lotus attempted to register separately the screen displays of 1-2-3 as an audiovisual work, the

Copyright Office denied the registration. In a letter to plaintiff, Copyright Examiner Julia Huff explained:

> textual screen displays embodied within the computer program that generates them are covered by the registration for the program, without either the need or justification for separate registration for the displays. Because the displays are considered to be an integral part of the program, the authorship in the displays appears to be the same as that contained in the program.

Letter of Jan. 20, 1987, from Huff to plaintiff (Tr. Ex. 1092).

Accordingly, I conclude that Lotus has properly registered 1-2-3 for copyright protection, and that this court does have subject matter jurisdiction over all aspects of this infringement suit.

B. Laches and Equitable Estoppel

Next, defendants argue that plaintiff is barred from any relief under the equitable doctrines of laches and equitable estoppel.

Defendants have the burden of proving laches. *Costello v. United States*, 365 U.S. 265, 282, 81 S.Ct. 534, 543, 5 L.Ed.2d 551 (1961). Thus, to prevail on this affirmative defense. defendants must prove by a preponderance of the evidence (1) that plaintiff inexcusably and unreasonably delayed in bringing this action; and (2) that the delay has prejudiced defendants. *Gardner v. Panama R. Co.*, 342 U.S. 29, 31, 72 S.Ct. 12, 13, 96 L.Ed. 31 (1951); *Puerto Rican-American Insurance Co. v. Benjamin Shipping Co.*, 829 F.2d 281, 283 (1st Cir.1987). Defendants have failed to sustain either part of that burden.

According to defendants, beginning as early as April 30, 1985, "the trade press trumpeted Paperback's soon-to-be-released spreadsheet product [VP-Planner], making no secret that it would look and work like Lotus 1-2-3." Defendants' Post-Trial Brief at 120. VP-Planner was eventually released on October 30, 1985, but, defendants' charge, Lotus waited until January 12, 1987, 438 days later, to bring suit. *Id.*

Even if Lotus did wait fourteen-and-a-half months after learning of VP-Planner's release before commencing this litigation, I find that this delay was neither unreasonable nor inexcusable. Defendants cite *Worcester Brewing Corp. v. Rueter & Co.*, 157 F. 217, 219 (1st Cir.1907), for the proposition that a fifteen-month delay barred plaintiff's trademark infringement damage claim. That case, though, involved a relatively simple claim that defendant infringed plaintiff's trademark "Sterling Ale." Here, as evidenced by the court's lengthy Opinion, plaintiff's claim of copyright infringement in nonliteral elements of the user interface of 1-2-3 was a far more complicated claim, and one that required much more time to evaluate and prepare.

In circumstances such as these, a delay of fourteen-and-a-half months is not only reasonable, but, in fact, is likely to serve both part and public interests in appropriate use of public and private resources for resolution of disputes. As plaintiff aptly asserts: "Prudent business judgment, Rule 11 and basic common sense required Lotus first to ascertain that the threat to its intellectual property interest was serious and that its legal position was sound before filing suit." Plaintiff's Post-Trial Brief at 83. *See* Manzi Decl. ¶¶ 8-11 (Docket No. 293); *Roulo v. Russ Berrie & Co., Inc.*, 886 F.2d 931, 942 (7th Cir.1989) (twenty-one-month delay in filing infringement suit reasonable where "delay in filing was due to [plaintiff's] inquiry into the facts to determine the merit of her claim against [defendant]"), *cert. denied*, ___ U.S. ___, 110 S.Ct. 1124, 107 L.Ed.2d 1030 (1990).

Defendants nevertheless argue that, during the fourteen-and-a-half-month delay, plaintiff at least had a duty to notify defendants that plaintiff considered VP-Planner objectionable and that plaintiff intended to claim proprietary rights in 1-2-3's user interface. Not only did plaintiff fail to so notify defendants, defendants assert, but. in fact, plaintiff affirmatively led defendants to infer that VP-Planner was not objectionable. Accordingly, defendants contend, plaintiff is estopped from recovering against defendant. Defendants' Post-Trial Brief at 131 n. 66 (*citing Precious Metals Associ-*

ates, Inc. v. Commodity Futures Trading Commission, 620 F.2d 900, 908 (1st Cir.1980) (" 'A person is estopped from denying the consequences of his conduct where the conduct has been such as to induce another to change his position in good faith or such that a reasonable man would rely upon the representations made.' ") (in turn *quoting Bergeron v. Mansour*, 152 F.2d 27, 30 (1st Cir.1945))).

In support of this contention, defendants cite a form letter that plaintiff's Supervisor of the Information Center sent to Paperback in January 1986 inquiring as to whether Lotus could use its copy of VP-Planner at any location at plaintiff's business. Tr. Ex. 201. Defendants argue that this letter "suggested that Lotus treated VP-Planner as it treated any other non-objectionable software product." Defendants Post-Trial Brief at 131.

First, defendants have failed to offer any evidence that they did, in fact, draw that inference. Indeed, Steven Cook, Paperback's former president, testified that the form letter was simply returned to plaintiff with the appropriate box checked. and that Paperback did not even retain a copy of the letter. Cook Affdvt. ¶ 24 (Docket No. 299). Second, the court is unable to fathom how any reasonable person could draw that inference from Lotus' form letter. In any event, defendants have cited no authority to support the proposition that Lotus' letter, and Lotus' failure to notify defendants of its internal investigation into possible legal action during the fourteen-and-a-half month hiatus, constitute the kind of actions that support laches or equitable estoppel.

Another reason that defendants have failed to sustain a laches defense is that they have failed to demonstrate, any prejudice.

At the core of defendants' asserted prejudice is the following contention advanced by Paperback's former president:

> If Lotus had sued Paperback within three or four months after the release of VP-Planner, then Paperback would not have continued to market, sell, or develop VP-Planner in its Lotus-compatible form, whether or not the litigation had any merit.

Cook Affdvt. ¶ 33.

In light of defendants' belief that their spreadsheet product, to be successful, had to be compatible with 1-2-3, and defendants' strident insistence that 1-2-3's user interface is not copyrightable, this contention is simply not credible. In hindsight, perhaps, the expense of this lawsuit looms immense, but the court does not credit the testimony that the mere threat of a lawsuit, especially one that defendants assert they deemed frivolous, would have caused defendants' to retreat from marketing a 1-2-3 "workalike" ' spreadsheet program. I find that defendants have not shown that they were prejudiced by any delay.

Accordingly, I find that defendants have failed to sustain either element of a laches defense, or an equitable estoppel defense.

IX. RESERVED RULINGS ON OBJECTIONS

Although ruling on most evidentiary objections during the trial, the court reserved ruling on some Rule 402 objections until after the court had an opportunity to consider the relevance of certain evidence in the context of the entire case. Most notable is evidence proffered by defendants with respect to VisiCalc, which plaintiff asserted was irrelevant to any issues raised by this case.

Although I agree that the "VisiCalc story," in the decisionmaking process reflected in this Opinion, has turned out to be not relevant to any of the equitable defenses, and not relevant to defendants' contention that Lotus itself copied aspects of VisiCalc with 1-2-3, I conclude that the "VisiCalc story" is relevant for background purposes. *See Granite Music Corp. v. United Artists Corp.*, 532 F.2d 718, 720-21 (9th Cir.1976) (admitting evidence of similar musical phrases appearing in works that predate both plaintiff's and defendant's song proper to show, for background purposes, that certain elements were of such ordinary and common occurrence that the probability of independent coincidental production was great). Also, I conclude that this evidence may appropriately be considered for the purpose of deciding premise

facts upon which the conclusions of law with respect to copyrightability rest. *See generally* Part VII(D), *supra*.

With respect to all other reserved evidentiary rulings, the court concludes that they are moot in view of the findings and rulings stated in this Opinion. Moreover, even if the court were to rule explicitly that all of the evidence was admissible, I find as factfinder that none of this evidence alone, nor all of it taken together, would change any of the findings and conclusions contained in this Opinion. That is, even when taken into account, this evidence has so little probative weight that it does not change any finding. Nevertheless, any party wishing an explicit ruling on any reserved evidentiary objection may so move on or before July 19, 1990, identifying specifically each objection on which a more explicit ruling is requested with references, if available, to specific pages in the transcript of the first phase of this trial.

ORDER

For the reasons explained in this Opinion and on the findings and conclusions stated, it is hereby ordered:

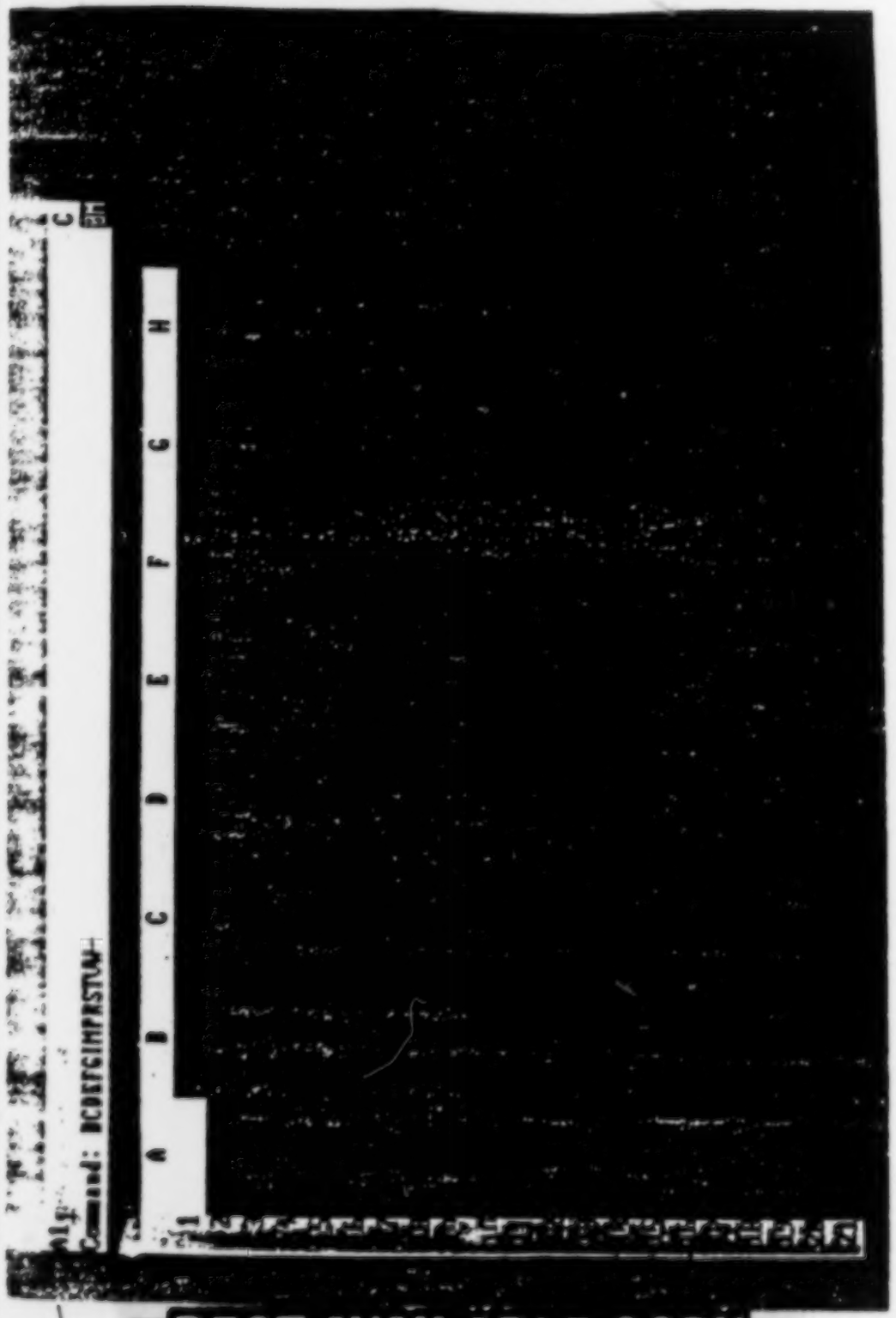
(1) Liability for infringement by defendants' is established.

(2) Any party wishing an explicit ruling on any reserved evidentiary objection may so move on or before July 19, 1990, identifying specifically each objection on which a more explicit ruling is requested with references, if available, to specific pages in the transcript of the first phase of this trial.

(3) A conference with respect to further proceedings will be held as specified in the Joint Procedural Order of this date.

APPENDIX 1

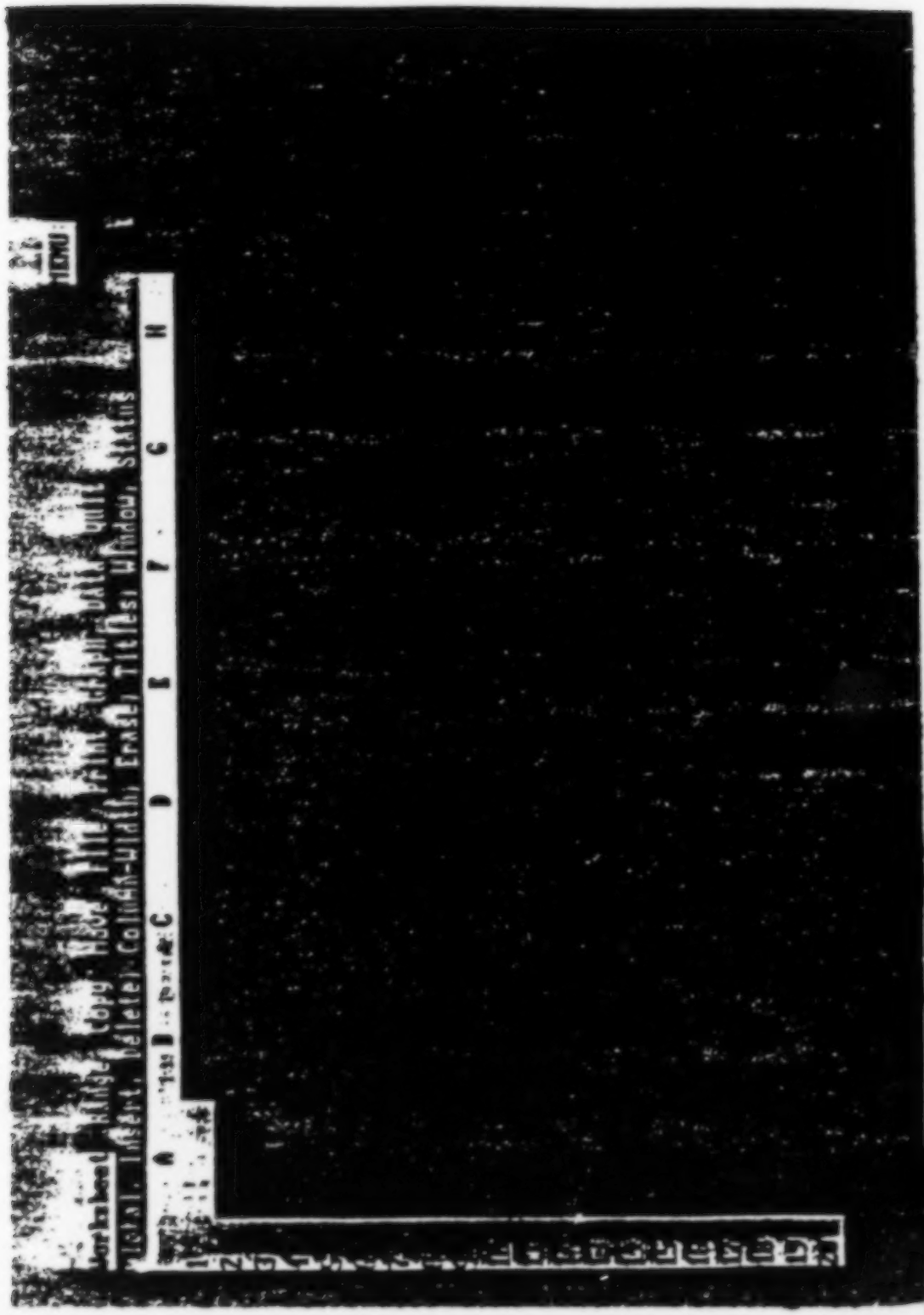
VisiCalc Screen, Main Menu
Tr. Ex. 1047, Tab A1



BEST AVAILABLE COPY

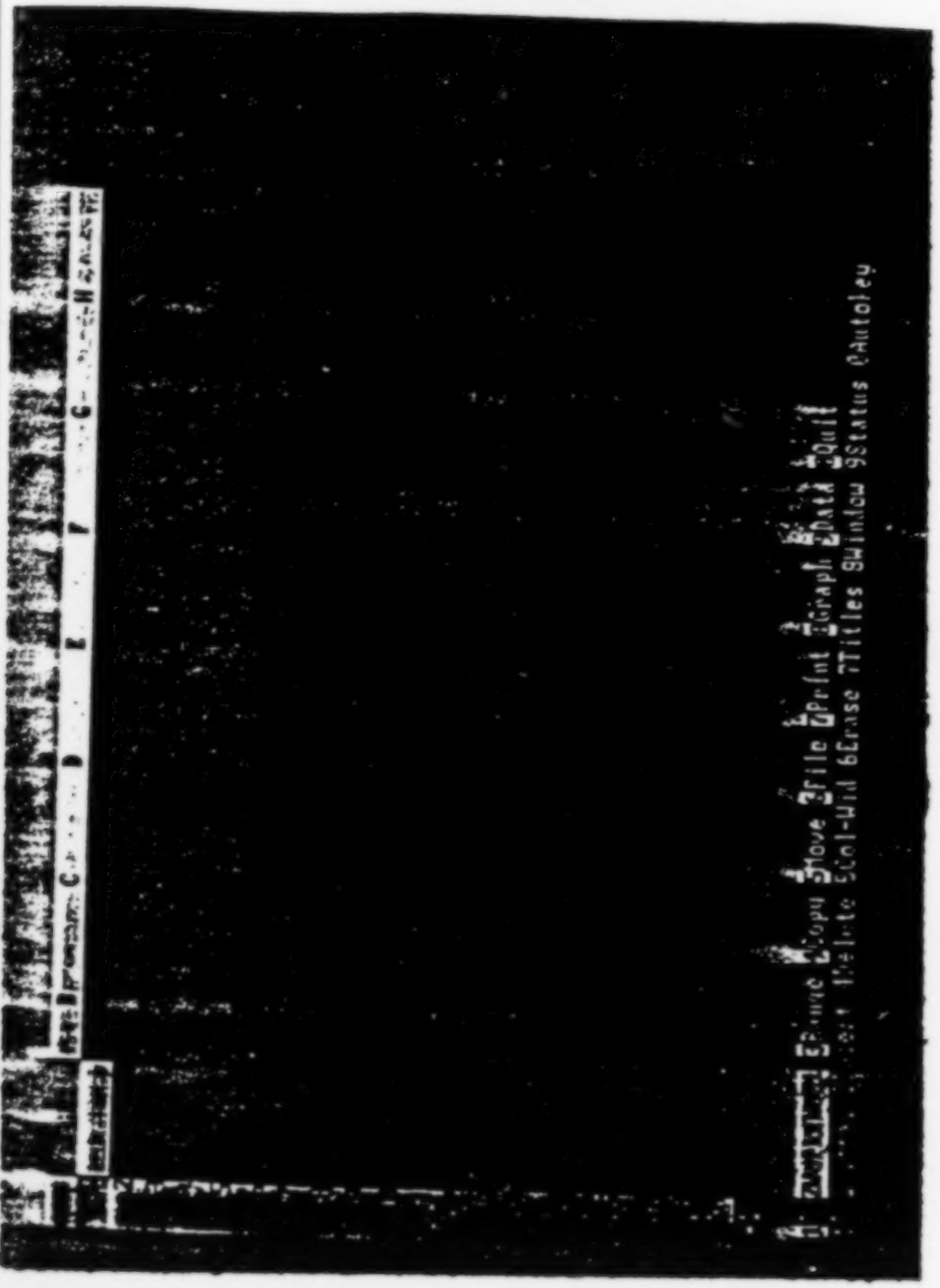
APPENDIX 2

1-2-3 Screen, Main Menu, Worksheet highlighted
Tr. Ex. 1047, Tab A2



APPENDIX 3

VP-Planner Screen, Main Menu, Worksheet highlighted
Tr. Ex. 1047, Tab A4



COPYRIGHT ACT

§ 101. Definitions

Except as otherwise provided in this title, as used in this title, the following terms and their variant forms mean the following:

An "anonymous work" is a work on the copies or phonorecords of which no natural person is identified as author.

An "architectural work" is the design of a building as embodied in any tangible medium of expression, including a building, architectural plans, or drawings. The work includes the overall form as well as the arrangement and composition of spaces and elements in the design, but does not include individual standard features.

"Audiovisual works" are works that consist of a series of related images which are intrinsically intended to be shown by the use of machines or devices such as projectors, viewers, or electronic equipment, together with accompanying sounds, if any, regardless of the nature of the material objects, such as films or tapes, in which the works are embodied.

The "Berne Convention" is the Convention for the Protection of Literary and Artistic Works, signed at Berne, Switzerland, on September 9, 1886, and all acts, protocols, and revisions thereto.*

---

* Section 2 of the Berne Convention Implementation Act of 1988, Pub. L. 100-568, 102 Stat. 2853 (Oct 31, 1988), provides the following declarations:

The Congress makes the following declarations:

(1) The Convention for the Protection of Literary and Artistic Works, signed at Berne, Switzerland, on September 9, 1886, and all acts, protocols, and revisions thereto (hereafter in this Act referred to as the "Berne Convention") are not self-executing under the Constitution and laws of the United States.

(2) The obligations of the United States under the Berne Convention may be performed only pursuant to appropriate domestic law.

(*footnote continued*)

A work is a "Berne Convention work" if—

(1) in the case of an unpublished work, one or more of the authors is a national of a nation adhering to the Berne Convention, or in the case of a published work, one or more of the authors is a national of a nation adhering to the Berne Convention on the date of first publication;

(2) the work was first published in a nation adhering to the Berne Convention, or was simultaneously first published in a nation adhering to the Berne Convention and in a foreign nation that does not adhere to the Berne Convention;

(3) in the case of an audiovisual work—

(A) if one or more of the authors is a legal entity, that author has its headquarters in a nation adhering to the Berne Convention; or

(B) if one or more of the authors is an individual, that author is domiciled, or has his or her habitual residence in, a nation adhering to the Berne Convention;

(4) in the case of a pictorial, graphic, or sculptural work that is incorporated in a building or other structure, the building or structure is located in a nation adhering to the Berne Convention; or

(5) in the case of an architectural work embodied in a building, such building is erected in a country adhering to the Berne Convention.

For purposes of paragraph (1), an author who is domiciled in or has his or her habitual residence in, a nation adhering to the Berne Convention is considered to be a national of that nation. For purposes of paragraph (2), a work is considered to

---

(3) The amendments made by this Act, together with the law as it exists on the date of the enactment of this Act, satisfy the obligations of the United States in adhering to the Berne Convention and no further rights or interests shall be recognized or created for that purpose.

have been simultaneously published in two or more nations if its dates of publication are within 30 days of one another.

The "best edition" of a work is the edition, published in the United States at any time before the date of deposit, that the Library of Congress determines to be most suitable for its purposes.

A person's "children" are that person's immediate offspring, whether legitimate or not, and any children legally adopted by that person.

A "collective work" is a work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole.

A "compilation" is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. The term "compilation" includes collective works.

A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

"Copies" are material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. The term "copies" includes the material object, other than a phonorecord, in which the work is first fixed.

"Copyright owner", with respect to any one of the exclusive rights comprised in a copyright, refers to the owner of that particular right.

The "country of origin" of a Berne Convention work, for purposes of section 411, is the United States if—

(1) in the case of a published work, the work is first published—

(A) in the United States;

(B) simultaneously in the United States and another nation or nations adhering to the Berne Convention, whose law grants a term of copyright protection that is the same as or longer than the term provided in the United States;

(C) simultaneously in the United States and a foreign nation that does not adhere to the Berne Convention; or

(D) in a foreign nation that does not adhere to the Berne Convention, and all of the authors of the work are nationals, domiciliaries, or habitual residents of, or in the case of an audiovisual work legal entities with headquarters in, the United States;

(2) in the case of an unpublished work, all the authors of the work are nationals, domiciliaries or habitual residents of the United States, or, in the case of an unpublished audiovisual work, all the authors are legal entities with headquarters in the United States; or

(3) in the case of a pictorial, graphic, or sculptural work incorporated in a building or structure, the building or structure is located in the United States.

For the purposes of section 411, the "country of origin" of any other Berne Convention work is not the United States.

A work is "created" when it is fixed in a copy or phonorecord for the first time; where a work is prepared over a period of time, the portion of it that has been fixed at any particular time constitutes the work as of that time, and where the work has been prepared in different versions, each version constitutes a separate work.

A "derivative work" is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a "derivative work".

A "device", "machine", or "process" is one now known or later developed.

To "display" a work means to show a copy of it, either directly or by means of a film, slide, television image, or any other device or process or, in the case of a motion picture or other audiovisual work, to show individual images nonsequentially.

A work is "fixed" in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration. A work consisting of sounds, images, or both, that are being transmitted, is "fixed" for purposes of this title if a fixation of the work is being made simultaneously with its transmission.

The terms "including" and "such as" are illustrative and not limitative.

A "joint work" is a work prepared by two or more authors with the intention that their contributions be merged into inseparable or interdependent parts of a unitary whole.

"Literary works" are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

"Motion pictures" are audiovisual works consisting of a series of related images which, when shown in succession, impart an impression of motion, together with accompanying sounds, if any.

To "perform" a work means to recite, render, play, dance, or act it, either directly or by means of any device or process or, in the case of a motion picture or other audiovisual work, to show its images in any sequence or to make the sounds accompanying it audible.

"Phonorecords" are material objects in which sounds, other than those accompanying a motion picture or other audiovisual work, are fixed by any method now known or later

developed, and from which the sounds can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. The term "phonorecords" includes the material object in which the sounds are first fixed.

"Pictorial, graphic, and sculptural works" include two-dimensional and three-dimensional works of fine, graphic, and applied art, photographs, prints and art reproductions, maps, globes, charts, diagrams, models, and technical drawings, including architectural plans. Such works shall include works of artistic craftsmanship insofar as their form but not their mechanical or utilitarian aspects are concerned; the design of a useful article, as defined in this section, shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.

A "pseudonymous work" is a work on the copies or phonorecords of which the author is identified under a fictitious name.

"Publication" is the distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending. The offering to distribute copies or phonorecords to a group of persons for purposes of further distribution, public performance, or public display, constitutes publication. A public performance or display of a work does not of itself constitute publication.

" 'Registration,' for purposes of sections 205(c)(2), 405, 406, 410(d), 411, 412, and 506(e), means a registration of a claim in the original or the renewed and extended term of copyright."

To perform or display a work "publicly" means—

> (1) to perform or display it at a place open to the public or at any place where a substantial number of persons outside of a normal circle of a family and its social acquaintances is gathered; or

(2) to transmit or otherwise communicate a performance or display of the work to a place specified by clause (1) or to the public, by means of any device or process, whether the members of the public capable of receiving the performance or display receive it in the same place or in separate places and at the same time or at different times.

"Sound recordings" are works that result from the fixation of a sense of musical, spoken, or other sounds, but not including the sounds accompanying a motion picture or other audiovisual work, regardless of the nature of the material objects, such as disks, tapes, or other phonorecords, in which they are embodied.

"State" includes the District of Columbia and the Commonwealth of Puerto Rico, and any territories to which this title is made applicable by an Act of Congress.

A "transfer of copyright ownership" is an assignment, mortgage, exclusive license, or any other conveyance, alienation, or hypothecation of a copyright or of any of the exclusive rights comprised in a copyright, whether or not it is limited in time or place of effect, but not including a nonexclusive license.

A "transmission program" is a body of material that, as an aggregate, has been produced for the sole purpose of transmission to the public in sequence and as a unit.

To "transmit" a performance or display is to communicate it by any device or process whereby images or sounds are received beyond the place from which they are sent.

The "United States", when used in a geographical sense, comprises the several States, the District of Columbia and the Commonwealth of Puerto Rico, and the organized territories under the jurisdiction of the United States Government.

A "useful article" is an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a "useful article".

The author's "widow" or "widower" is the author's surviving spouse under the law of the author's domicile at the time of his or her death, whether or not the spouse has later remarried.

A "work of visual art" is—

(1) a painting, drawing, print, or sculpture, existing in a single copy, in a limited edition of 200 copies or fewer that are signed and consecutively numbered by the author, or, in the case of a sculpture, in multiple cast, carved, or fabricated sculptures of 200 or fewer that are consecutively numbered by the author and bear the signature or other identifying mark of the author, or

(2) a still photographic image produced for exhibition purposes only, existing in a single copy that is signed by the author, or in a limited edition of 200 copies or fewer that are signed and consecutively numbered by the author.

A work of visual art does not include—

(A)(i) any poster, map, globe, chart, technical drawing, diagram, model, applied art, motion picture or other audiovisual work, book, magazine, newspaper, periodical, data base, electronic information service, electronic publication, or similar publication; (ii) any merchandising item or advertising, promotional, descriptive, covering, or packaging material or container; (iii) any portion or part of any item described in clause (i) or (ii);

(B) any work made for hire; or

(C) any work not subject to copyright protection under this title.

A "work of the United States Government" is a work prepared by an officer or employee of the United States Government as part of that person's official duties.

A "work made for hire" is—

(1) a work prepared by an employee within the scope of his or her employment; or

(2) a work specially ordered or commissioned for use as a contribution to a collective work, as a part of a motion picture or other audiovisual work, as a translation, as a supplementary work, as a compilation, as an instructional text, as a test, as answer material for a test, or as an atlas, if the parties expressly agree in a written instrument signed by them that the work shall be considered a work made for hire. For the purpose of the foregoing sentence, a "supplementary work" is a work prepared for publication as a secondary adjunct to a work by another author for the purpose or introducing, concluding, illustrating, explaining, revising, commenting upon, or assisting in the use of the other work, such as forewords, afterwords, pictorial illustrations, maps, charts, tables, editorial notes, musical arrangements, answer material for tests, bibliographies, appendixes, and indexes, and an "instructional text" is a literary, pictorial, or graphic work prepared for publication and with the purpose of use in systematic instructional activities.

(As amended Dec. 12, 1980, Pub. L. 96-517, § 10(a), 94 Stat. 3028; Oct. 31, 1988, Pub. L. 100-568, § 4(a)(1), 102 Stat. 2854-55; Dec. 1, 1990, Pub. L. 101-650, Title VII, § 703, 104 Stat. 5133; June 26, 1992, Pub. L. 102-307, Title I, § 102, 106 Stat. 266.); Oct. 28, 1992, Pub. L. 102-563, § 3, 106 Stat. 4248.)

§ 102. Subject matter of copyright: In general

(a) Copyright protection subsists, in accordance with this title in original works or authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories:

(1) literary works;

(2) musical works, including any accompanying words;

(3) dramatic works, including any accompanying music;

(4) pantomimes and choreographic works;

(5) pictorial, graphic, and sculptural works;

(6) motion pictures and other audiovisual works;

(7) sound recordings; and

(8) architectural works.

(b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained illustrated, or embodied in such work.

(As amended Dec. 1, 1990, Pub. L. 101-650, Title VII, § 703, 104 Stat. 5133.)

§ 103. Subject matter of copyright: Compilations and derivative works

(a) The subject matter of copyright as specified by section 102 includes compilations and derivative works, but protection for a work employing preexisting material in which copyright subsists does not extend to any part of the work in which such material has been used unlawfully.

(b) The copyright in a compilation or derivative work extends only to the material contributed by the author of such work, as distinguished from the preexisting material employed in the work, and does not imply any exclusive right in the preexisting material. The copyright in such work is independent of, and does not affect or enlarge the scope, duration, ownership, or subsistence of, any copyright protection in the preexisting material.

§ 117. Limitations on exclusive rights: Computer programs*

Notwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

(As amended Dec. 12, 1980, P.L. 96-517, § 10(b), 94 Stat. 3028.

---

* Section 117 as amended by the Act of Dec. 12, 1980, Pub. L. 96-517 (94 Stat. 3015). Prior to this amendment, Section 117 read as follows:

§ 117. Scope of exclusive rights: Use in conjunction with computers and similar information systems

Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law, whether title 17 or the common law or statutes of a State, in effect on December 31, 1977, as held applicable and construed by a court in an action brought under this title.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 87-0076-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

PAPERBACK SOFTWARE INTERNATIONAL
and STEPHENSON SOFTWARE, LTD.,

Defendants.

Civil Action No. 87-0074-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

MOSAIC SOFTWARE, INC.,

Defendant.

AFFIDAVIT OF MITCHELL D. KAPOR

MITCHELL D. KAPOR, being duly sworn, deposes and says:

1. My name is Mitchell D. Kapor and I reside in Brookline, Massachusetts.

2. I make this affidavit to provide testimony in the Phase I trial of the above-captioned actions.

3. I am the Chairman of the Board and Chief Executive Officer of ON Technology, Inc.

* * *

[*][1]

8. In November 1977, I purchased my first personal computer, a Radio Shack TRS-80, and began writing computer programs for use with that computer.

9. In July 1978 I bought an Apple II personal computer and became an independent software consultant, specializing in developing software for the Apple computer.

10. In early 1979 I formed Micro Finance Systems ("MFS"), a computer software development company.

11. By 1979 I had created, designed and written a computer program called Tiny Troll, the first graphics and statistics software package for the Apple II personal computer. Tiny Troll was published by MFS.

12. In 1979 I also enrolled in the Accelerated Masters Program at the Sloan School of Management at MIT, eventually completing three of the requisite four terms of the program.

13. In the fall of 1979, while at the Sloan School, I was approached by the principals of Personal Software, Inc. ("PSI") to inquire whether MFS was interested in creating a software program for PSI to publish, market and sell. I agreed

[1] Bracketed asterisks indicate beginning of page in original document.

to do so. (In addition, during the brief period from approximately April to August of 1980 I was employed by PSI in California).

14. Pursuant to a software distribution agreement between MFS and PSI dated November 3, 1979, I created, designed and wrote two computer software programs, known as VisiPlot and VisiTrend, which were first published, marketed and released for [*] sale by PSI in the spring of 1981. VisiPlot was a graphics program and VisiTrend was a statistical analysis and forecasting program.

15. Through the Data Interchange Format ("DIF"), a file format and data exchange protocol that I helped design (and which was eventually released to the public by Software Arts, Inc.), users of VisiCalc (a spreadsheet product), VisiPlot and VisiTrend could transfer data between those programs. For example, data stored in files created using VisiCalc could be displayed graphically by transferring the data to, and then using, the VisiPlot program.

16. In the fall of 1981, the software distribution agreement between MFS and PSI was terminated. I sold all rights to the VisiPlot and VisiTrend products to PSI pursuant to an Agreement and Bill of Sale. (A true and correct copy of the Agreement and Bill of Sale is attached as Exhibit A to this affidavit.)

17. In the Agreement and Bill of Sale, I explicitly retained rights to two of my own product concepts, "EBS" and "I-C-L language system and programs". EBS, or Executive Briefing System, a business/professional graphics program, later became Lotus' first product. I-C-L, a concept for an integrated software package that would include spreadsheet, graphics and other capabilities, served as the foundation for a long process of development resulting in the eventual creation of Lotus 1-2-3.

[*]

18. In January 1982, I founded Professional Software Technology, which in April 1982 was combined with Lotus

Development Company (another company I had created) to form Lotus Development Corporation.

The Conceptual Origins of Lotus 1-2-3

19. The original idea for Lotus 1-2-3 came to me in early 1981, while I was developing VisiPlot and VisiTrend. In the course of using DIF to transfer data from VisiCalc to VisiPlot—a cumbersome process—it occurred to me that a product which, unlike these independent, single-purpose programs, would combine spreadsheet and graphing capabilities in a single, integrated program, could have significant market appeal.

20. In the early summer of 1981, I met with Jonathan Sachs ("Sachs"), who was developing various electronic spreadsheet programs. That June, Sachs joined MFS with what I understood to be non-exclusive rights to one of his spreadsheet programs.

21. After Sachs joined MFS, it became our mutual goal and purpose to create an integrated business software product for a personal computer which would be centrally organized around the spreadsheet metaphor, and which would be superior to existing products.

22. The result of our efforts to attain that goal, Lotus 1-2-3, Release 1.0, was first commercially shipped (or "released") on January 26, 1983.

[*]

23. Throughout the preceding eighteen months, Sachs and I considered hundreds of alternative ideas for virtually every aspect of the product. 1-2-3 ultimately evolved from a careful selection process wherein concepts were often expressed in one form, later disappeared, and then reappeared in another guise—either in 1-2-3 itself or, frequently, in a subsequent Lotus product.

24. The ebb and flow of this continual selection process is reflected in thousands of pages of contemporaneous development notes and documents from my files. True and correct copies of some of these notes, handwritten by me during this development period, are attached to this affidavit as Exhibits B through K. These notes are from files that have been maintained, by me or my attorneys since their creation. It was my usual practice during that time to write on the face of my notes the date of their creation.

25. From the outset, I intended to create a product that would have value for, and appeal to, the broadest possible audience. I wanted the product to attract not only experienced computer programmers or users, but also individuals who had no prior computer or spreadsheet experience.

26. To accomplish this, I determined that it was essential to create a product with a user interface dramatically different from those provided in then-existing spreadsheet products—one that would be easy for the inexperienced to learn and [*] to use, while also providing speed, convenience, and advanced capabilities for the more sophisticated.

27. As my notes of June 18, 1981 (Exhibit B) reflect, early in the development process of 1-2-3 I perceived the user interface as something distinct and separable from the functionality that the spreadsheet program would provide to the user—*i.e.*, the set of operations and functions that the user could perform with the software.

28. Indeed, one of my earliest concepts was to provide a user interface which would permit the user to define his or her own menus of commands, corresponding to the elements of the program's underlying functionality that he or she desired to use. This would, in effect, allow each user to create an individually customized user interface, even if the user had no knowledge of the workings of the program's underlying source code.

29. I had conceived this idea previously in connection with my work on VisiPlot, as my notes of December 29, 1980 (Exhibit C) reflect.

30. This concept, which I came to refer to as a User Interface Generator ("UIG"), is reflected in my notes of March 3 and May 29, 1981 (Exhibits D and E respectively).

Prototype Development

31. From the summer of 1981 through the end of the year, Sachs and I worked on a prototype for a spreadsheet which

* * *

[*]

previous concept for allowing users also to define their own menus. (The concept reemerged later in modified fashion.)

40. I determined that a fixed menu structure was more suitable for the novice spreadsheet users to whom, in large part, we sought to appeal, because it would provide a framework in which such users could learn to approach the solution of spreadsheet problems generally. A fixed menu structure would also help to guide the user through the program's capabilities and to select appropriate operations to solve specific application problems.

41. My early thought to include user-defined menus did survive in one feature of the published version of 1-2-3, expressed as the "/XM" command, which allows experienced users to create their own alternative menus.

42. The fixed menu structure I chose to employ incorporated the two-line moving cursor menu (a concept I first introduced in VisiPlot), which permits the invocation of menu commands from predetermined arrays of choices, and which also provides information about those choices to the user, serving in that respect as a kind of on-line reference.

43. The main elements of the two-line moving cursor menu are: (a) words representing a set of available command options displayed in the top line; (b) long prompts displayed in the second, or lower line, containing information concerning the commands displayed in the top line; and (c) the abil-

ity to [*] review, by moving the cursor along the words in the top line, the information concerning each command option in that menu before selecting any command from that menu.

44. I decided to organize these menus hierarchically, meaning that the selection of one command option from the first level menu could lead in turn to another array of command options on a second level menu (or "submenu"), and so on, branching out in a sequence of descending menu levels. This organization enables the novice user to browse through the menu levels, in order to view the valid sequences of available options (and their corresponding explanations) and to map out a plan for performing a particular task.

45. This hierarchical organization of command menus can be described, or visualized, as a "menu tree". Indeed, the menu structure of Lotus 1-2-3 as released is depicted in the accompanying user's manual in an index entitled, "1-2-3 and PrintGraph Command Trees".

46. The choice of words selected to represent commands and their organization in the menu tree, I believed, would be critical to the success of the user interface, especially for the novice user. I believed that the words should not only inform, but also fit into the overall approach to spreadsheet problems reflected in the menu tree. I also believed that the flow of the menu tree—*i.e.*, its internal logic, moving from level to level—should be easy for the user to understand and reflect as [*] closely as possible the user's natural way of thinking, so that the user would never have to think like a computer.

47. These beliefs were, in part, derived from my familiarity with previous spreadsheet products, which I felt were deficient in this regard. For example, the alphabet-like menu structures of some existing products, such as VisiCalc, conveyed too little information to the user, required too much memorization and/or cross-referencing to the printed user's manual, and—because they were not, to a meaningful extent, arranged hierarchically—failed to provide the user with much insight into an effective approach to the solution of spread-

68. Despite my decision to employ only a fixed menu structure for the product as a whole, during this time I continued to refine my ideas for user-definable menus. For many months I considered including a "command language" in the product, which would enable the user to automate repetitive command sequences. This idea ultimately resulted in the development and incorporation in 1-2-3 of keystroke "macros". A macro permits a user to store a sequence of keystrokes which the user can later instruct the computer to execute by invoking a single command.

69. I was aware of the concept of macros from prior experiences with text editor programs. To my knowledge, no other spreadsheet at the time expressed the macro idea the way we did in 1-2-3.

70. Although there was much give-and-take on the desirability of including macro capabilities, after lengthy experimentation I ultimately decided to do so in the belief that sophisticated users would find them valuable.

71. We also considered the selection of commands to provide to the user for construction and operation of macros. The universe of possible commands was very broad, some of which could present considerable complexity. Ultimately, we determind to include a set of eight commands that we believed would be relatively easy for users of some sophistication to grasp, but [*] would also enhance the functionality of the macro feature to a meaningful degree.

Development of 1-2-3; Menu Structure

72. The essential functionality the product would provide was settled approximately in August 1982. Our primary focus then turned to the precise iteration of the menu tree.

73. From August until about October 1982, I met with Sachs and the other members of our development team, including John Posner, Ezra Gottheil, Mary Goldschmidt and Steve Miller, several times a week to determine each word choice, location and menu arrangement.

74. On my own and collectively with this team, hundreds of hours were spent refining the menus, and dozens of iterations of menus and their organization were proposed. The process induced intense debate at times. In the end, the final decisions were mine.

75. In selecting words for the menus, we struggled to choose words that would intelligently convey to the user the purpose of each command and its underlying functionality, while at the same time ensuring that each word in a particular menu level began with a unique letter. The first concern was based on our desire to make the menus as informative and intuitive as possible. The latter concern derived from the previous design choice to permit the user to invoke menu commands by striking the key corresponding to the first letter of each word; obviously, if [*] more than one word began with the same first letter, the program could not determine which command the user intended by striking the key corresponding to both.

76. Another factor we considered in organizing the menus was my belief—based upon the principle of cognitive psychology that individuals can retain only seven pieces of information (plus or minus two) at a time—that each menu level should contain approximately seven items (or fewer), to be easier to learn and use.

77. I also believed that the hierarchical arrangement of the menus should reflect a structured approach that communicated the product's underlying functionality to the user. For example, I felt that commands representing related features should be grouped together on a single level of the menu tree, with subsequent dependent commands appearing on lower levels of the tree.

78. In addition, in organizing the menus I believed that the commands most likely to be used frequently should be grouped in a broad menu appearing at the top of the menu structure. Similarly, I felt that the commands displayed within a given menu level should appear from left to right in declining order of likely frequency of usage.

79. Because there was no precedent for the menu structure we envisioned, we had no formal market research or other data to guide us in making these determinations. We did, [*] however, conduct a number of informal user tests both internally at Lotus as well as outside the company. In the end, the menu tree organization was based largely on my intuition and subjective judgment, informed by feedback from the user tests, trying as best I could to imagine myself in the rule of a typical user.

80. The numerous possibilities considered in creating the final Lotus 1-2-3 menu structure are exemplified in my contemporaneous notes of May 29, July 24, and August 18, 1982 (Exhibits I, J, and K respectively), which depict the proposed menu structure at various stages of development.

81. Attached as Exhibits L through O to this affidavit are true and correct copies of documents maintained in my files created on or about August 4, August 17, September 22, and October 4, 1982, respectively, which I recognize as reflecting the Lotus 1-2-3 menu structure as it existed on or about those dates.

82. As Exhibits I through O reflect, there were several possible choices for almost every word selected, as well as for its location in the menu trees. For example, the "Copy" command could also have been called "Replicate", "Duplicate", "Repeat", or "Clone".

83. One example of an instance where both the word used to represent a command, and its location in the menu tree, underwent considerable change in the course of development is that of the "Range" command and the commands branching from it.

84. Its origin was an idea I had in April 1982 to [*] include a "named range" feature, whereby the user could refer to ranges of cells in the worksheet by assigning them user-defined names (in addition to referencing them by other means). I thought this feature, not present in any existing product, would be useful for specifying commonly referred-to

ranges on the worksheet, including tables, data to be graphed, and areas to be printed.

85. I originally contemplated implementing this feature through a command called "Name" located on the top menu level, which, when invoked, would display all named ranges the user had previously specified horizontally in the control panel. A user could select a desired named range by moving the cursor to the applicable name and striking the "Enter" key, which would then invoke a submenu containing the following options: Accept, Exit, Edit, Rename, Delete, and Cancel.

86. Throughout the summer I struggled to simplify and refine the "named range" feature. One alternative I considered was to assign a "Name" command to a function key, either instead of or in addition to a "Name" menu command.

87. Finally, following numerous meetings and discussions, in August I decided to replace "Name" in the top menu level with "Range", and to locate a "Name" menu command—permitting the user to create, modify or delete named ranges—in the branching submenu. The feature permitting the user to select previously-specified named ranges from a horizontal display was [*] assigned to a "Name" function key. In September, the command options for the submenu branching further from the "Range Name" commands were selected.

88. During the intensive phase of work on the menus from August to October 1982, another debate erupted regarding the "Worksheet Global" command options. These commands were intended to permit the user to set certain parameters for what we called the "worksheet", including the format, recalculation mode, recalculation order and current and default column widths.

89. These commands initially branched from a command called "Set" located in the top menu level. I became dissatisfied with this structure because I did not believe it gave sufficient emphasis to the "global" or worksheet-wide nature of most of the commands; at the same time, some of the com-

mands following "Set" were not, originally, of that same "global" nature.

90. My initial thought was to replace the "Set" command with a "Global" command option in the top menu. That, however, created an initial-letter conflict because the word "Graph" already appeared in that menu.

91. I had long discussions with my development team regarding alternative words to represent the "Global" command. One possibility suggested was "Default". Again, however, we already had chosen the word "Data" to represent the database branches of the menu tree in the top menu.

[*]

92. I ultimately decided that "Global" was the most descriptive and appropriate word. To resolve the initial-letter conflict with "Graph", I moved "Global" from the top menu to a submenu branching form the "Worksheet" command.

93. A related decision was to reorganize the features permitting the user to set current and default (or worksheet-wide) column widths, both of which originally branched form the "Global" command. I felt this was potentially confusing to the user. The feature allowing the user to set the width solely for the column he or she was then working in (the "current" column) did not logically fit with the rest of the "Global" commands.

94. To provide greater clarity and to present a better-structured approach in the menu tree, I broke these features apart, placing one "Column-Width" command in the submenu branching form "Global", to represent the capability to alter widths for all columns in the worksheet, and another "Column-Width" command in the same submenu as "Global", to represent the feature pertaining to the current column only. That way, the user would be unlikely to mistake the effect of a column-width adjustment, since the user would have to select the "Global" command only in instances when the user desired to make a global change.

95. It also became necessary to change the sequence of the "Global" command submenus as Sachs implemented new [*] functionality. For example, when Sachs initially added the "Iteration" and "Recalculation" features, they both appeared in the submenu directly branching from the "Worksheet Global" commands. Because of a shortage of space on the screen in that menu level, the word "Recalculation" was abbreviated to "Recalc".

96. Ultimately, however, as Sachs continued to refine the program's functionality, we decided to move "Iteration" down one menu level in the sequence, to branch from the "Recalc" command. Once this change was made and additional space was available, I restored "Recalc" to its original form—"Recalculation".

97. We also confronted difficulty in choosing a word to represent the feature that permitted a user to save a selected range of cells (including formulas), rather than an entire worksheet, in a separate file. Originally, and throughout the summer of 1982, this feature was represented by the term "File Extract".

98. In September, when we changed the "File Delete" command option to "File Erase" (to avoid the conflict that arose when we renamed the "File Current" command "File Disk" and, ultimately, "File Directory"), we encountered yet another initial-letter conflict, since it was not possible to have both "Erase" and "Extract" on the same menu level. We could not find an acceptable alternative for "Erase"; thus, we had to reconsider "Extract".

[*]

99. One suggestion was "Partial". As is reflected in Exhibit N, I did not think "Partial" was a satisfactory alternative, because it was not sufficiently informative. One of the members of the team suggested using the non-English word "Xtract", so we could convey the desired information without conflicting keystrokes. "Xtract" is, in fact, the alternative that we ultimately used.

100. In addition to selecting the words to represent each command and determining their organization in the menu tree, from August to October 1982 we also created the corresponding long prompts displayed in the lower line of the menus. Each of the above-mentioned team members participated.

101. In some instances, we wrote brief textual explanations for the commands appearing in the menus. In others, we chose to display the next submenu branching from a particular command. Our decisions in this regard were governed not by a mechanical principle but by a case-by-case consideration as to which type of long prompt would be appropriate to a particular command, with the ultimate goal of providing better information to the user.

* * *

[*]

112. In September 1985, Lotus released a new version or upgrade of Lotus 1-2-3, called Release 2.0. I was involved in the development of Release 2.0 primarily in the roles of product design and project management and oversight.

113. Release 2.0's enhancements to the features provided in Release 1A resulted in the addition of a number of commands, and necessitated some changes to the existing menus. For example, the addition of the "Value" and "Transpose" commands to the submenu branching from the "Range" command required the abbreviation of the words "Label-Prefix" appearing in Release 1A to the single word "Label" in Release 2.0, to represent the same command. The addition of a "Hidden" command to the submenu branching from the "Range Format" commands led to a slight change in the order of items within that submenu.

114. However, the user interface of 1-2-3, Release 1A, expanded and modified as necessary to express new capabilities, was otherwise essentially incorporated in Release 2.0.

Signed under the penalties of perjury this 27th day of November, 1989.

/s/ MITCHELL D. KAPOR
Mitchell D. Kapor

OPPOSITION

BRIEF

(4)

No. 94-2003

Supreme Court, U.S.
FILED
JUL 6 1995
CLERK

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF IN OPPOSITION TO PETITION FOR CERTIORARI

GARY L. REBACK
(Counsel of Record)
MICHAEL BARCLAY
ISABELLA E. FU
WILSON, SONSINI,
GOODRICH & ROSATI
Professional Corporation
650 Page Mill Road
Palo Alto, California 94304
Telephone: (415) 493-9300

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY &
GELHAAR
One Post Office Square
33rd Floor
Boston, Massachusetts 02109
Telephone: (617) 542-7400

Attorneys for Respondent

BOWNE OF SAN FRANCISCO, INC. • 345 BANSOME ST. • S.F., CA 94104 • (415) 362-2300

BEST AVAILABLE COPY

(3844)

QUESTION PRESENTED

Did the First Circuit correctly hold that commands used to operate a computer spreadsheet program are uncopyrightable under 17 U.S.C. § 102(b)?

RULE 29.1 STATEMENT

Respondent Borland International, Inc. has no parent corporation or subsidiaries that are not wholly owned, except for certain foreign subsidiaries in which a minimal amount of shares (fewer than 1%), which are not publicly traded, are held by foreign nationals in accordance with local law.

TABLE OF CONTENTS

	Page
QUESTION PRESENTED	i
RULE 29.1 STATEMENT	ii
TABLE OF AUTHORITIES	v
INTRODUCTION AND SUMMARY	1
STATEMENT OF THE CASE	7
A. Factual Background	8
1. The Lotus Product	8
2. The Borland Product	11
B. Proceedings in the District Court	11
REASONS FOR DENYING THE PETITION	13
I. THE CIRCUITS ARE IN AGREEMENT THAT WORDS USED AS BUTTONS TO OPERATE A PROGRAM, UNLIKE THE PROGRAM ITSELF, ARE UNCOPYRIGHTABLE	13
A. The District Court Erroneously Applied the "Abstractions" Test to Uncopyrightable Subject Matter	13
B. The District Court's Decision to Extend Copyright to the Words That Operate the Program Ran Afoul of Both the Statute and the Case Law	17

TABLE OF CONTENTS (Cont'd)

Page

C. There Is No Conflict Among the Circuits That Menu Commands and Similar Methods of Operation Are Uncopyrightable 20

II. THE FIRST CIRCUIT OPINION REMOVES THE UNCERTAINTY PRODUCED BY THE DISTRICT COURT AND UPHOLDS THE OVERALL INTELLECTUAL PROPERTY FRAMEWORK ESTABLISHED BY CONGRESS 24

A. The First Circuit Opinion Restores Clarity and Predictability to the Law 24

B. Both Copyright and Patent Remain Sources of Strong Protection, Providing Enormous Incentives for Authors and Inventors of Software Products to Innovate 26

CONCLUSION 27

TABLE OF AUTHORITIES

Page(s)

CASES

Ashton-Tate Corp. v. Ross,
728 F. Supp. 597 (N.D. Cal. 1989) 21

Ashton-Tate Corp. v. Ross,
916 F.2d 516 (9th Cir. 1990) 6, 20, 21, 24

Autoskill, Inc. v. National Educ. Support Sys., Inc.,
994 F.2d 1476 (10th Cir.), *cert. denied*,
114 U.S. 307 (1993) 21, 22

Baker v. Selden, 101 U.S. 99 (1879)..... *passim*

Brown Bag Software v. Symantec Corp.,
960 F.2d 1465 (9th Cir.), *cert. denied sub. nom.*
BB Asset Mgmt. Inc. v. Symantec Corp.,
113 S. Ct. 198 (1992) 20

Computer Assoc. Int'l, Inc. v. Altai, Inc.,
982 F.2d 693 (2d Cir. 1992) ..... *passim*

Digital Comm. Assoc., Inc. v. Softklone Distrib. Corp.,
659 F. Supp. 449 (N.D. Ga. 1987) 22

Engineering Dynamics, Inc. v. Structural Software, Inc.,
26 F.3d 1335 (5th Cir. 1994) 7, 23

Engineering Dynamics, Inc. v. Structural Software, Inc.,
46 F.3d 408 (5th Cir. 1995) 7, 23

Gates Rubber Co. v. Bando Chem. Indus., Ltd.,
9 F.3d 823 (10th Cir. 1993) 6, 15, 16, 22

TABLE OF AUTHORITIES (Cont'd)

Page(s)

Gates Rubber Co. v. Bando Chem. Indus., Ltd.,
No. 92-S-136 (D. Colo. filed June 12, 1995) ... 6, 22

Johnson Controls, Inc. v. Phoenix Control Sys., Inc.,
886 F.2d 1173 (9th Cir. 1989) 20

Lotus Dev. Corp. v. Borland Int'l, Inc.,
788 F. Supp. 78 (D. Mass. 1992) ..... *passim*

Lotus Dev. Corp. v. Borland Int'l, Inc.,
799 F. Supp. 203 (D. Mass. 1992) ..... *passim*

Lotus Dev. Corp. v. Borland Int'l, Inc.,
831 F. Supp. 202 (D. Mass. 1993) 2, 10

Lotus Dev. Corp. v. Borland Int'l, Inc.,
831 F. Supp. 223 (D. Mass. 1993) 2, 10

Lotus Dev. Corp. v. Borland Int'l, Inc.,
49 F.3d 807 (1st Cir. 1995) ..... *passim*

Lotus Dev. Corp. v. Paperback Software Int'l,
740 F. Supp. 37 (D. Mass. 1990) ..... *passim*

Manufacturers Technologies, Inc. v. CAMS, Inc.,
706 F. Supp. 984 (D. Conn. 1989) 22

Mitek Holdings, Inc. v. Arce Eng'g Co.,
864 F. Supp. 1568 (S.D. Fla. 1994) 23

Sony Corp. v. Universal City Studios, Inc.,
464 U.S. 417 (1984) 27

TABLE OF AUTHORITIES (Cont'd)

Page(s)

Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc.,
797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 14, 15, 23, 24

STATUTES, RULES, AND REGULATIONS

60 Fed. Reg. 28,778 (proposed June 2, 1995) 27

H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54-55
(1976), *reprinted in* 1976 U.S.C.C.A.N. 5659 ... 14

17 U.S.C. § 101 13

17 U.S.C. § 102(b) ..... *passim*

28 U.S.C. § 1292(a) 12

MISCELLANEOUS

Copyright Office Circular 61, *Copyright Registration for Computer Programs* 25

Paul Goldstein, *Copyright* (1989) 14, 25

Paul Goldstein, *Infringement of Copyright in Computer Programs*, 47 U. Pitt. L. Rev. 1119 (1986) 25

Karen S. Kovach, Comment, *Computer Software Design; User Interface — Idea or Expression?*
60 U. Cin. L. Rev. 161 (1991) 3

TABLE OF AUTHORITIES (Cont'd)

	Page(s)
Kenneth A. Liebman, *et al., Back To Basics: A Critique Of The Emerging Judicial Analysis Of The Outer Limits Of Computer Program Expression*, 2 Computer Law., Dec. 1985, at 1 ..	15
Steven W. Lundberg et al., *Identifying Uncopyrightable Computer Implemented Processes and Systems*, 9 Computer Law., Apr. 1992, at 7	3
David Nimmer & Melville B. Nimmer, *Nimmer on Copyright* (1994)	13
Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback*, 55 Law & Contemp. Probs., Spring 1992, at 311	3
Timothy S. Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 Stan. L. Rev. 1061 (1993)	3
U.S. Department of Commerce, Press Release No. 95-18, *USPTO to Develop Guidelines to Protect Software Inventions*, March 30, 1995	27
Julian Velasco, *The Copyrightability of Nonliteral Elements of Computer Programs*, 94 Columbia L. Rev. 242 (1994)	3

TABLE OF CONVENTIONS

Borland	—	Respondent Borland International, Inc.
Lotus	—	Petitioner Lotus Development Corporation
Petition or Pet.	—	Lotus' Petition for a Writ of Certiorari filed in this Court
Pet. App.	—	Appendix to Lotus' Petition filed in this Court
1st Cir. App.	—	Joint Appendix of Record filed in the U.S. Court of Appeals for the First Circuit (the court below)
Borland S.J. Brief	—	Borland's Memorandum in Support of Cross-Motion for Summary Judgment (Dkt. No. 141) in the district court
Ex.	—	Exhibit ____ from Exhibits in Support of Borland's Cross-Motion for Summary Judgment (Dkt. No. 142) in the district court
RE	—	Excerpts of Record attached to Borland's opening brief in the court below (filed Dec. 14, 1993)

No. 94-2003

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF IN OPPOSITION TO PETITION FOR CERTIORARI

INTRODUCTION AND SUMMARY

The Court of Appeals in this case held that the commands used to operate a computer spreadsheet program — common commands such as "COPY," "MOVE," or "PRINT" displayed to the user on a computer screen—are uncopyrightable under § 102(b) of the Copyright Act. The court's opinion relied on the express language of the statute that forbids copyright protection for "methods of operation" and "systems." The court also relied on this Court's seminal decision in the area, *Baker v. Selden*, 101 U.S. 99 (1879), which mandates that claims for the protection of methods of operation and systems be governed by the *patent* law rather than the *copyright* law.

The First Circuit's opinion carefully reviewed the leading authority from the other circuits, principally the Second Circuit's decision in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), that sets forth a methodology for evaluating claims of copyright infringement in the text (*i.e.*, "code") and "structure" of computer programs. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 814 (1st Cir. 1995), Pet. App. at 13a–15a. The First Circuit readily agreed that the methodology used in *Altai* and similar cases provides "a useful framework" for evaluating infringement claims in "code" and "code structure" but was simply inapposite to the issue here—the copyrightability of command words, or "menus," used to operate the program. 49 F.3d at 815, Pet. App. at 14a. The First Circuit viewed this issue as easily resolvable by reference to the statute and Supreme Court authority.

Judge Boudin filed a separate opinion in which he concurred in the majority's reasoning as well as its conclusion. 49 F.3d at 821, Pet. App. at 27a–28a. Judge Boudin went on to explain that extending copyright protection to the command words at issue *by judicial fiat*, as the district court had done, is at variance with the intent of Congress and is both inefficient and anticompetitive from an economic perspective. Lotus should look to the patent law, rather than copyright, to protect its method of operation.

The First Circuit decision reversed a series of opinions in this case from a single judge in the District of Massachusetts.[1] From the beginning, the district court eschewed a narrow focus on what was actually at issue in this case—the method by which the user tells the computer program what to do. Instead, the district court

---

[1] *See Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78 (D. Mass. 1992) ("*Borland I*"), Pet. App. at 145a; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) ("*Borland II*"), Pet. App. at 106a; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 202 (D. Mass. 1993) ("*Borland III*"), Pet. App. at 71a; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223 (D. Mass. 1993) ("*Borland IV*"), Pet. App. at 29a. Each of these opinions refers to and is based upon an earlier decision of the district court, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990) ("*Paperback*"), Pet. App. at 183a, which decision was not appealed.

viewed the case as the opportunity to involve itself in the developing case law regarding the protection of computer program "code" and "structure." Although those issues are *not* present in this case, the district court sought to inject this case into the debate about those issues, by creating its own novel methodology to determine the copyrightability of all aspects of computer technology.

The Lotus petition describes the district court opinions variously as "virtually canonical," as having "a prominent role" and as a "touchstone" in computer copyright law. Petition at 2, 15, 22. Exactly the opposite is true. The district court decisions in this case and in the predecessor *Paperback* case provoked a firestorm of controversy. As the record below reflects, the district court opinions were widely criticized in the academic community,[2] the legal press,[3] the national financial press,[4] and the computer

---

[2] *See, e.g.*, Steven W. Lundberg et al., *Identifying Uncopyrightable Computer Implemented Processes and Systems*, 9 Computer Law., Apr. 1992, at 7, 9 ("the Court in *Lotus* [*v. Paperback*] could never have reached the correct conclusion since it never laid the fundamental groundwork for it"); Timothy S. Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 Stan. L. Rev. 1061 (1993); Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback*, 55 Law & Contemp. Probs., Spring 1992, at 311, 352-53; Karen S. Kovach, Comment, *Computer Software Design: User Interface—Idea or Expression?* 60 U. Cin. L. Rev. 161 (1991) (*Paperback* improperly extended copyright protection to Lotus' menu command system); Julian Velasco, *The Copyrightability of Nonliteral Elements of Computer Programs*, 94 Columbia L. Rev. 242, 262–65 (1994). These authorities are cited in Borland's Petition for Initial In Banc Hearing, filed with the First Circuit on November 10, 1993 ("Borland's In Banc Brief"), at 2 n.9.

[3] *See, e.g.*, Susan Kostal, *Copyright Scholars Want a Fine Point Put on Spreadsheet Case*, S.F. Daily J., Oct. 4, 1991, at 1, 8; *Lotus Wins Copyright Suit*, Mass. Law. Wkly., Aug. 10, 1992, at 23. *See* Borland's In Banc Brief at 3 n.10.

[4] *See, e.g., Borland Gains in Bid For Appeals Ruling On Lotus Copyrights*, Wall. St. J., Aug. 20, 1993, at B3; William M. Bulkeley,

industry press[5] for extending copyright protection to put large sectors of the software industry off limits to competition, thereby producing a radical departure from the leading authority of other circuits. Although Lotus claims that the Second Circuit's *Altai* decision cites the district court opinions at issue here "with approval," Pet. at 16, in fact the Second Circuit *specifically cited* and pointedly *rejected* the district court's "incentive-based arguments in favor of broad copyright protection" as having a "corrosive effect on certain fundamental tenets of copyright doctrine." 982 F.2d at 712. Astonishingly, the Lotus petition fails to disclose the Second Circuit's pointed criticism of the district court's position.

In the proceedings below, a myriad of industry associations (representing both large and small companies), consumer groups, distinguished academics, eminent computer scientists, and even the Register of Copyrights of the United States all filed amicus briefs criticizing the district court and urging reversal of its opinions.[6] Lotus' position, by contrast, has attracted little support

---

Borland Case Briefs Raise Questions About Software Copyright Protection, Wall. St. J., Oct. 4, 1991, at B4; William M. Bulkeley, *Borland Loses Infringement Claim by Lotus*, Wall St. J., Aug. 13, 1993, at B5; T.R. Reid, *Consumers Lose When Software Makers Sue Each Other*, Wash. Post, Oct. 25, 1993, at F18; John R. Wilke, *Ruling Against Borland May Intensify Copyright Debate*, Wall. St. J., August 3, 1992, at B1, B4. See Borland's In Banc Brief at 3 n.11.

[5]Heather Clancy, *Copyright laws blurred*, Computer Reseller News, Aug. 17, 1992, at 137; Peter Coffee, *Key Reader verdict shows need for open standards*, PC Week, Aug. 30, 1993, at 32; Ed Foster, *That's a nice macro you wrote. Too bad Lotus now owns it*, InfoWorld, Aug. 23, 1993, at 45; Ed Foster, *Too-tight legal controls can straightjacket a whole industry*, InfoWorld, Sept. 6, 1993, at 49; Beth Freedman, *Experts Question Ruling In Lotus-Borland Case*, PC Week, Aug. 10, 1992, at 6; Steve Gibson, *Lotus ruling will damage the industry if it protects languages*, InfoWorld, Sept. 14, 1992, at 82. *See* Borland's In Banc Brief at 3 n.12.

[6]*See* the following: Brief Amicus Curiae of Computer Software Industry Association, representing over 3000 companies and professional individuals; Brief of Amicus Curiae Software Entrepreneurs' Forum,

in the *software* industry. Its amici have consisted of a few large companies who favor the extension of copyright law to inhibit competition from the hundreds of other operating system and application software companies.

Lotus' argument that the First Circuit decision created a conflict in the circuits is pure fabrication. Lotus argues that there is a consensus in the circuits regarding the application of the "abstractions" test to computer programs, that the district court's opinions in this case are a part of that consensus, and that the First Circuit's decision departs from that body of law. But, in fact, the discussion in the circuit courts regarding the "abstractions" test is directed to different subject matter—*i.e.*, the program code and structure—rather than the menus used to operate the program. That is the principal failing in Lotus' argument and the reason Lotus lost this case. As the First Circuit made plain, this case is *not* about a computer program; it is about the menu words that are used as buttons and switches to operate the program. The First Circuit decision does not stand for the proposition that it is error to apply the "abstractions" test *to computer programs*. Rather, the First Circuit has held that it is error to apply the "abstractions" test to something (in this case menus) that is not copyrightable in the first place.

Nor did the First Circuit reject the "idea/expression" dichotomy in applying § 102(b) as Lotus asserts. Pet. at 23. Rather, referring to the express language of the statute, the First Circuit recognized that § 102(b) makes "methods of operation" and "systems" as well as "ideas" uncopyrightable. The First Circuit applied the dichotomy in the statute and case law, finding that the commands are on the uncopyrightable side of the line (which

---

representing over 1000 independent software developers, consultants and software providers; Amicus Brief of 81 Distinguished Computer Scientists; Brief of Amicus Curiae (PC User Groups), representing over 16,500 individual and corporate PC users, including Fortune 500 companies; Brief of Amicus Curiae of American Committee for Interoperable Systems (including members such as Sun Microsystems, Tandem Computers and AT&T Global Information Solutions); Two Briefs Amicus Curiae on behalf of 25 Distinguished Copyright Law Professors; and Brief of Amicus Curiae on behalf of the Register of Copyrights.

Lotus calls the "distinction") between "ideas," "methods of operation," and "systems," on the one hand, which are uncopyrightable, and "expression" and "description," on the other hand, which are copyrightable.[7]

Contrary to Lotus' position, *no circuit* has held that the menu commands that operate the program are copyrightable on *any theory*, "idea/expression" dichotomy, "abstractions" or otherwise. In fact, it was the district court in this case that departed from existing authority when it declined to follow the Ninth Circuit's holding that the menu command hierarchy of a spreadsheet is uncopyrightable under 17 U.S.C. § 102(b). In declining to follow the Ninth Circuit's decision, *Ashton-Tate Corp. v. Ross*, 916 F.2d 516, 521–22 (9th Cir. 1990), the district court stated:

> In the interest of completeness and candor, I note as well that courts in one circuit are not bound by the decisions of other circuits.

Borland II, 799 F. Supp. at 220, Pet. App. at 136a. Apparently lacking the "completeness and candor" of the district court, the Lotus petition does not even mention the *Ashton-Tate* decision.

The only other circuit (the Tenth Circuit) to have directly considered the copyrightability of menu commands vacated a district court decision holding that menu commands are copyrightable and provided instructions for further consideration of the issue. *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 843–44 (10th Cir. 1993). On remand, the district court held that the menu commands were *uncopyrightable*, citing the First Circuit's *Lotus* decision approvingly. *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, No. 92-S-136 (D. Colo. filed June 12, 1995).

---

[7] It is true, of course, that a computer program might be described as a "method of operating" the computer and, as a consequence, Congress was required to expressly include "computer programs" in the copyright statute to ensure that computer programs would be copyrightable. But Congress did not amend the copyright statute to make menu commands, buttons, switches, and similar methods of operating a program copyrightable. On the contrary, § 102(b) of the Copyright Act plainly states that "methods of operation" are uncopyrightable. Lotus would have the courts make law that contradicts the express language of the statute.

Lotus' assertion that "[i]t is now legal to copy menus in the First Circuit but not in the Fifth, Ninth or Tenth," Pet. at 28, is simply preposterous.

Similarly, the Lotus petition claims that a particular Fifth Circuit decision "relied heavily" on the district court's analysis. Pet. at 23, *citing Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1348–49 (5th Cir. 1994). Incredibly, the Lotus petition fails to point out that following publication of its initial opinion, the Fifth Circuit was deluged by requests for rehearing from software and semiconductor companies, computer scientists and user groups, and subsequently *issued a supplemental opinion*. The Fifth Circuit limited its first decision to the facts of that case and specifically disclaimed reliance on the rationale that underlies the district court's opinions at issue here. *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408, 409 (5th Cir. 1995). It is unfathomable that Lotus could assert that neither the Second Circuit nor the Fifth, Ninth and Tenth Circuits "has suggested that the central tenets of [the district court's] analysis have been anything but correct." Pet. at 22.

In short, there is simply no conflict among the circuits on the protection of menu commands and similar methods used to operate computer programs. Furthermore, given the subject matter that is really at issue in this case and the narrow basis of the First Circuit's opinion, virtually any comment by this Court on the copyrightability of code or program structure in a review of this case would be *dicta*. There are, on the other hand, a variety of cases currently proceeding through the lower courts that directly address those issues (discussed in Section IC below).

STATEMENT OF THE CASE

There is good reason why the district court's opinions in this case generated such widespread attention and controversy. In the usual software copyright case, the defendant is alleged to have copied either the text ("code") or structure of the plaintiff's computer program, or the way the plaintiff's program looks on the computer screen when it is executing. No such copying occurred here. Lotus did not even allege any copying of its code or code

structure, and even the district court found that the programs looked different on the computer screen. *Borland II*, 799 F. Supp. at 220, Pet. App. at 137a.

Users operate the program at issue here through a series of commands—common English words—arranged in hierarchies called "menus." 49 F.3d at 809, Pet. App. at 4a. As the First Circuit opinion explains, the precise facts of this case are critical to its disposition. The menu commands of the product at issue do not function as labels or explanations for any buttons or switches used to operate the program. Rather, the menu commands *themselves* are used to operate the program in much the same way that the buttons on a video cassette recorder (VCR) operate that machine. While such utilitarian methods of operation are perhaps patentable under the patent laws, this is not a patent case since Lotus obtained no patent on this or any other relevant aspect of Lotus 1-2-3.

As the First Circuit found, "the entire Lotus menu command hierarchy is essential to operating" the program. 49 F.3d at 815–16, Pet. App. at 15a–22a. If a user types "COPY" or "C," the program copies. Typing "PRINT" or "P" causes the computer to print. Typing more complex sequences of commands in the hierarchy executes other operations of the spreadsheet. There is other text attendant to a computer program—not copied by Borland—that communicates with the user or that provides information to the user. Such text includes books about the program, manuals, on-screen "help" text, and other textual material that *does* not operate the computer. In contrast, the words of the Lotus menu command hierarchy are literally the method of operating the spreadsheet program. These limiting facts are crucial to the disposition of this petition.

A. Factual Background

1. *The Lotus Product*

The development of the Lotus menu command hierarchy is well chronicled in the Lotus petition. According to the affidavit

submitted with the petition[8] the commands were organized "hierarchically," and the manner depicted in a "menu tree," so that "the selection of one command option from the first level menu could lead in turn to another array of command options on a second level menu (or 'submenu') and so on." Kapor Aff. at ¶¶44-45, Pet. App. at 287a. The Lotus developer drew a firm line between the menu choices in the hierarchy (the "command options") and the explanation of those command options. According to the affidavit, for example, the "long prompts" (or "screen help" text) are intended to provide "information to the user" and "explanations," while the menu commands are directed toward "performing a particular task." Kapor Aff. at ¶¶44, 101, Pet. App. at 287a, 296a.

As set forth in great detail in the lower court record, the Lotus spreadsheet was a great success, but that success had little to do with the menu command hierarchy. Rather, when the IBM PC was introduced in August 1981, the Lotus developers, according to the district court, "exploited this opportunity" by designing the technical aspects of their spreadsheet product to take advantage of the technological advances of the IBM PC over pre-existing computers. *Paperback*, 740 F. Supp. at 65-66, Pet. App. at 231a.

The menu command hierarchy was simply not a qualitatively significant part of the product *at the time of its introduction*,

---

[8]It should be noted that the Kapor Affidavit was *not* prepared in support of Lotus' position in the *Borland* case. Rather, the affidavit was prepared for Lotus in the earlier *Paperback* case in which the defendant copied Lotus' entire screen display, not merely the command words. Subsequent to the *Paperback* decision, as the record below indicates, the author of the affidavit (who is the principal designer of the Lotus product) made it clear in Congressional testimony that he does not believe it is beneficial to the industry to extend copyright protection to individual elements of a screen display—*e.g.*, the menus. That, in his view, would constitute "overprotection" that is "pernicious" and "stifling." See Brief of Amicus Curiae Software Entrepreneur's Forum, filed in the First Circuit on Dec. 23, 1993, at 6.

either from Lotus' viewpoint or the viewpoint of users.[9] The exact words and order of the Lotus menu command hierarchy were not important to the product's initial success, but they *became* vitally important to the success of later versions of Lotus' product and spreadsheets offered by Lotus' competitors for two reasons. First, as Judge Boudin explains (and the record below demonstrates in detail), users invested their own time, money and energy in learning the Lotus commands as keystroke combinations to operate the spreadsheet, just as users operate a typewriter to produce readable text by touch typing on the standard QWERTY keyboard. 49 F.3d at 819-21, Pet. App. at 24a-26a. Second, as both the majority opinion and Judge Boudin explain,[10] users automate those steps by creating "macros," computer programs written by users themselves.

Lotus' own documents and the record below demonstrate that while Lotus' product initially became a success because it was technologically superior to its early competition, it later maintained its share because, as Judge Boudin noted, the user's investment in learning the method of operation of the Lotus product and the creation of macros "locked in" those users who first selected Lotus over its inferior early competition.[11] Therefore, unless a new entrant with a superior product in the spreadsheet market could compete for the business of the vast majority of computer users who initially chose Lotus, competition would be limited solely to new spreadsheet users, a minor portion of the market. In short, there would be little, if any, business for which to compete.

---

[9] *See, e.g.*, Raburn Decl. at ¶14, RE 3: *Borland III*, 831 F. Supp. at 213, Pet. App. at 86a-87a.

[10] 49 F.3d at 809-10, 819-21, Pet. App. at 4a, 24a-27a. *See also Paperback*, 740 F. Supp. at 64-65, Pet. App. at 228a; *Borland II*, 799 F. Supp. at 213-14, Pet. App. at 110a; *Borland IV*, 831 F. Supp. at 227, Pet. App. at 31a.

[11] Ex. 38 (Houdini Analysis of Competitive Products) at L047694, 1st Cir. App. 1196. As a result of macros, 1-2-3 became, in the words of Lotus' own documents, "entrenched." Ex. 39 (Review of Excel) at L046265, 1st Cir. App. 1198.

2. *The Borland Product*

The Lotus petition attempts to dismiss Borland's product as a "clone" or imitator, but the record below demonstrates otherwise. The Borland product was first introduced in 1989 and won every major award for spreadsheet excellence given in the software industry. The Borland product invariably ranked higher than the Lotus product in head-to-head reviews and user comparisons, including those conducted by Lotus.[12]

Not only was the Borland product superior from technological and performance perspectives, but it also employed a new screen display that was different in every respect (save the command words) from the Lotus products at issue in this case. *See* Borland S.J. Brief (Dkt. No. 141) at 116–34. The Borland product had its own menu command hierarchy designed specifically to take advantage of its superior technological features and a different screen display. But the Borland product also provided, as an alternative, an enhanced version of the Lotus menu command hierarchy so that users who were locked into the earlier menus could operate the new product without relearning their spreadsheet skill set or rewriting their macros.[13]

B. Proceedings in the District Court

Throughout the proceedings below, the district court resisted every attempt Borland made to secure prompt appellate review of the district court's controversial extension of copyright law. The district court's earlier *Paperback* decision had not been appealed because Lotus settled the case following the district court's ruling and the defendant went out of business.

The district court issued its first opinion in this case on March 20, 1992, indicating that it intended to continue to employ its widely criticized methodology from the *Paperback* case. *Borland I*, 788 F. Supp. at 89-90, Pet. App. at 163a. On July 31,

---

[12] *See* Ex. 1 and 2, 1st Cir. App. 1134; Borland S.J. Brief (Dkt. No. 141) at 2 n.3.

[13] P. Kahn Tr. at 65–66, RE 10–11; Bosworth Tr. at 148, 1st Cir. App. 748; Ex. 27 at L034481, 1st Cir. App. 1178.

1992, the district court published its *Borland II* decision and an accompanying procedural order holding that "[t]he menu commands and menu hierarchy of Lotus 1-2-3 have expressive aspects and are copyrightable." Order Regulating Jury Trial, finalized Sept. 30, 1992 (Dkt. No. 232), at 18. However, the district court said that it could not determine the "precise scope of Borland's infringement" without further trial proceedings, *Borland II*, 799 F. Supp. at 221, Pet. App. at 138a, and set a trial to begin on November 2, 1992.

Immediately following the district court's *Borland II* decision, Borland pulled the Lotus menus from its products. On three separate occasions, Borland moved to certify for interlocutory appeal the question of the menu command hierarchy's copyrightability. However, the district court denied all three of Borland's motions, greatly extending the proceedings.[14] Ultimately, Borland argued in favor of enjoining its own product so that appellate review would be available pursuant to the injunction statute, 28 U.S.C. § 1292(a). On August 19, 1993, the district court entered a permanent injunction against the continued sale of Borland's product and an appeal was promptly taken to the First Circuit.

The district court refused to stay either damages discovery or a damages trial pending the First Circuit's decision on liability. As a result, the damages phase of the case, which was eventually terminated by the First Circuit's reversal, cost Borland millions of dollars in legal and expert fees. Although the district court was eventually reversed, the series of district court opinions, followed by the entry of a permanent injunction, had a devastating effect on Borland. For example, as the record below reflects, Borland's stock, which traded at 86¾ in January, 1992, prior to the first of the district court's opinions in the case, fell to 12¾ in September 1993, following entry of the injunction. Borland was required of financial necessity to sell its spreadsheet products to another company and is no longer a competitor of Lotus, which is now owned by International Business Machines Corp. (IBM). Any

[14] *See* Tr. of Sept. 23, 1992 Hearing at 54, 1st Cir. App. 18; Tr. of Oct. 16, 1992 Hearing at 15-17, 1st Cir. App. 24-25; Tr. of Aug. 19, 1993 Hearing (Dkt. No. 406) at 42-43.

further appellate proceedings, even if successful from Borland's legal viewpoint, can be exploited by IBM to further damage Borland competitively.

REASONS FOR DENYING THE PETITION

I. THE CIRCUITS ARE IN AGREEMENT THAT WORDS USED AS BUTTONS TO OPERATE A PROGRAM, UNLIKE THE PROGRAM ITSELF, ARE UNCOPYRIGHTABLE.

A. The District Court Erroneously Applied the "Abstractions" Test to Uncopyrightable Subject Matter.

The Lotus petition attempts to inject this case into the ongoing debate in the case law involving the extent to which the "structure" of a computer program should be protected by copyright. But extending copyright protection to menus and similar methods of operation has far greater legal and economic consequences than extending copyright to the structure of a computer program. The legal argument that at least some aspects of a program's structure might be copyrightable is a familiar one. A computer program, as defined in 17 U.S.C. § 101 (a "set of instructions" used "to bring about a certain result"), is a "literary work." The literal elements of the program, the source code and object code, are copyrightable in the same way that the literal text of a play would be.

Within the genre of *artistic* literary works like plays, courts have created the "paraphrasing" doctrine, *see* Pet. at 10, also known as the concept of "comprehensive non-literal similarity," as a basis for copyright infringement. *See* David Nimmer & Melville B. Nimmer, *Nimmer on Copyright*, § 13.03[A][1] at 13-29 (1994). If, for example, someone made an unauthorized translation into French of a play originally written in English, the authors would have a claim for copyright infringement, notwithstanding the fact that the original work (in English) and the unauthorized "copy" (in French) do not share any common literal expression. They are not substantially similar in terms of literal expression, but they are similar with respect to the "struc-

ture" of the play, each of its acts, each of its scenes, and, for that matter, the breakdown of the dialogue into sentences.

Because Congress decided to apply copyright protection to the code of a computer program (the "set of instructions"), one might argue (as Lotus does) that copyright doctrines for artistic literary works should apply to some extent to the code of a computer program, which is also a literary work. *See, e.g., Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1233–34 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987). Thus, if a programmer writes a program in "BASIC" and someone "translates" that program into the programming language "C," the second program is an unauthorized copy notwithstanding the absence of literal similarity at the code level. The two programs are "substantially similar" with respect to their detailed "structure." Hence, a few courts initially applied the doctrine of "comprehensive non-literal similarity" to protect the "structure, sequence and organization" of a computer program, at least to some limited extent. *See, e.g., Altai*, 982 F.2d at 702–04 (citing cases).

But application of the concept of "comprehensive non-literal similarity" to computer programs has been controversial because, unlike artistic works (such as plays), computer programs are utilitarian objects—they perform a function.[15] Copyright, unlike patent, is a very broad, long-lasting, easily obtainable type of protection, and Congress has always taken care to ensure that easily obtainable copyright protection would not be available for the utilitarian or functional aspects of works.[16] As Professor Miller, formerly a member of CONTU, stated:

> The end purpose of a computer program is to achieve a utilitarian result, *i.e.*, the computer's performance of logical operations in a way that produces the desired practical consequence. One cannot compare, therefore, the underlying processes of a computer program with, say, the underlying plot structure of a novel or a screenplay of a movie. This, of course, is the distinction recognized by the Supreme Court long ago in the seminal decision of *Baker v. Selden*, 101 U.S. 99 (1879).

Kenneth A. Liebman, *et al., Back To Basics: A Critique Of The Emerging Judicial Analysis Of The Outer Limits Of Computer Program "Expression,"* 2 Computer Law., December 1985 at 1, 8 (quoting Arthur Miller). Professor Miller, now Lotus' counsel, summarized this concern best when he opined in a declaration in another case: "The creativity, ideas and utilitarian aspects of a copyrighted work must look elsewhere for legal protection."[17]

The earliest attempt to apply "non-literal similarity" to computer programs, the Third Circuit's *Whelan* decision, resulted in protection for virtually all of the program's structure. *Whelan*, 797 F.2d at 1238. The Second Circuit in *Altai* rejected the *Whelan* test because the *Whelan* test made too much copyrightable. *Altai*, 982 F.2d at 705-06. Under the *Whelan* test, the "function" of the computer program is the work's protectable idea, and "everything that is not necessary to that purpose or function would be part of the [protectable] expression of the idea." *Whelan*, 797 F.2d at 1236. Under the "abstractions" test formulated by the *Altai* court, far fewer aspects of code structure are protected by copyright. The leading post-*Altai* decision, the Tenth Circuit's *Gates Rubber* case, limited the protection of code structure even further by requiring the lower courts specifically to consider the proscriptions of § 102(b) in applying the abstractions and filtrations formulations. *Gates Rubber*, 9 F.3d at 833, 836.

The test formulated by the district court in this case was based on *Whelan* rather than *Altai*, and, for that reason, the *Altai* court

---

[15] *Altai*, 982 F.2d at 704; Paul Goldstein, *Copyright*, § 2.15 at 195 (1989) (*"Goldstein Treatise"*).

[16] *See, e.g.*, H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54-55 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667-68.

[17] In declarations in an earlier case in which Professor Miller was the retained expert for the party accused of infringement, Professor Miller opined on the CONTU deliberations and the role of copyright in protecting computer programs. Those declarations can be found in the record below as exhibits to the Reply Brief of Defendant/Appellant Borland International, Inc., filed in the First Circuit on March 1, 1994. The quoted portion in the text is from the second declaration at ¶29.

rejected the lower court's approach as having a "corrosive effect on certain fundamental tenets of copyright doctrine." *Altai*, 982 F.2d at 712. More germane for the purposes of Lotus' petition is the fact that the district court sought to apply an abstractions-type analysis to a menu command hierarchy which, unlike a "computer program," is not copyrightable in the first place. *Borland II*, 799 F. Supp. at 216-19, Pet. App. at 128a-135a.

The "abstractions" test was initially applied to the text of plays—clearly copyrightable subject matter. Similarly, the leading cases cited by Lotus—*Whelan, Altai* and *Gates Rubber*—apply the abstractions test to the code of "computer programs" which Congress expressly has said is copyrightable. By contrast, the district court here applied its own variant of the abstractions test to the method of operating the program, the menu commands. The Lotus petition deftly slides over this critical distinction. *Compare* Pet. at 14-16 (describing the application of the test by various courts of appeals to the "program") *with* Pet. at 17 (discussing the district court's application of the test to so-called non-literal elements).

As the First Circuit opinion points out, the abstractions test assumes that the work at issue is copyrightable. 49 F.3d at 815, Pet. App. at 14a. Application of the "abstractions" test to the menu command hierarchy inevitably led the district court to find *something* in the menu command hierarchy copyrightable, *id.*, and the district court concluded that the specific commands and order chosen by Lotus were protected by copyright. *Borland II*, 799 F. Supp. at 217, Pet. App. at 131a. In other words, as the First Circuit explained, the district court's test devolved to a question of whether choices exist for the subject matter at issue. 49 F.3d at 811, 816, Pet. App. at 6a, 17a. Under the district court's methodology, if there are choices, the subject matter at issue is copyrightable. But the mere existence of a choice does not turn uncopyrightable subject matter into copyrightable "expression." One might equally argue that the engines of a Ferrari and a Volkswagen embody different "expressions" of the process of internal combustion, or that the QWERTY and Dvorak keyboards are different ways of "expressing" the means by which the user operates a typewriter.

Lotus responds to this reasoning by arguing that "computer programs," unlike car engines and keyboards, are copyrightable. Pet. at 29. Had the district court applied its abstractions test to the *computer program*, Lotus' argument would be germane. But the district court applied its test to the menu command hierarchy, not to the program. *Borland II*, 799 F. Supp. at 216-19, Pet. App. at 128a-135a.

Had the district court applied an abstractions test to the *programs* at issue, it would have quickly found no code or "structural" similarity on any level whatsoever — nonliteral or otherwise. Indeed, there was no allegation in this case that Borland copied or even had access to the "structure" of the Lotus program. Lotus did not produce its code in discovery, neither party introduced the code of its program into evidence at trial, and there is every reason to believe that the Lotus program and the Borland program have vastly different structures to perform the same functions. In short, even after application of the "abstractions" test to the program, the district court would still be faced with the issue of whether the menu commands are copyrightable. That issue can only be resolved by reference to the statute and case law dealing with utilitarian works. In any event, the issues of nonliteral similarity and copyrightability of code structure could not properly be reviewed by this Court in this case, since the narrow issue present here does not raise those broader issues.

B. The District Court's Decision to Extend Copyright to the Words That Operate the Program Ran Afoul of Both the Statute and the Case Law.

The First Circuit's reversal of the district court rests on the familiar language of § 102(b):

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

Although the Lotus petition at one point claims that the First Circuit's opinion rejected the idea/expression dichotomy, Pet. at

23, the earlier discussion in the Lotus petition correctly points out that § 102(b) is, in fact, the legislative *embodiment* of the idea/expression dichotomy. Pet. at 10. In short, the First Circuit embraced the demarcation in § 102(b) between copyrightable subject matter and uncopyrightable "methods of operation," "systems," and "ideas" which must look to patent law for protection. It was the district court that altered that fundamental demarcation. The First Circuit merely followed the "line" between copyrightable and uncopyrightable subject matter previously established by Congress and this Court in *Baker v. Selden*.

Confronted with the plain language of § 102(b) proscribing copyright protection for methods of operation, the district court limited the statute to abstractions. In the view of the district court, whenever any words are attached to § 102(b) subject matter (such as "processes" or "methods of operation"), the words become copyrightable. *See Borland I*, 788 F. Supp. at 91, Pet. App. at 167a; 49 F.3d at 816, Pet. App. at 17a. As the First Circuit pointed out, limiting uncopyrightable § 102(b) subject matter to abstractions moves the line established long ago in *Baker v. Selden* between copyrightable and uncopyrightable subject matter. For more than 100 years, until the district court's opinion, copyright law was grounded on the proposition that the barest words that state a system or operate a machine (*e.g.*, "move," "copy"), as opposed to a description of those operations (*e.g.*, the long prompts) are uncopyrightable. The district court's opinions moved the "line" between copyrightable and uncopyrightable subject matter established in *Baker v. Selden*, and that is what produced the enormous public outcry.

In *Baker*, the plaintiff, Charles Selden, obtained a copyright on a pamphlet that explained a systematic approach to bookkeeping. The pamphlet contained a complex series of ledgers or forms, like the various screen displays in the Lotus user interface. The Selden forms each contained grids, columns, and various alternative short textual descriptive "headings" or "captions" (such as "Balance Forward") like the Lotus menu commands.

The defendant, Baker, published forms similar in headings and arrangement to those of Selden. Selden sued Baker for copyright infringement because of the similarity, arguing—as Lotus argues

here—that there was "original expression" in the selection, ordering and arrangement of the headings and columns of the ledgers each contained in his copyrighted pamphlet. *See Baker v. Selden*, 101 U.S. at 101.

Manifestly, since the words on Baker's forms were different from those on Selden's, this Court could have found for Baker solely on the ground that the textual labels were not substantially similar. But this Court did not take that route. Stating the principal issue in the case as whether Baker could use "similar ruled lines and headings, or ruled lines and headings made and arranged on substantially the same system, without violating [Selden's] copyright," *id.*, at 101, this Court held that Selden's ledgers, including their column arrangement and textual headings, were not copyrightable at all—and could be copied verbatim. *Id.* at 107.

As the *Altai* court observed, 982 F.2d at 704, the holding of *Baker* that methods of operations and systems are not copyrightable is *not* restricted to *pure abstractions*:

> [T]he holding in *Baker* goes farther. The [Supreme] Court concluded that those aspects of a work, which "must necessarily be used as incident to" the idea, system or process that the work describes, are also not copyrightable. 101 U.S. at 104.

The First Circuit relied upon *Baker* in the same manner as did *Altai*. It relied upon Baker's limitations on the scope of copyright to conclude that the commands used to operate the Lotus program were not copyrightable. 49 F.3d at 816–17, Pet. App. at 18a. Indeed, the district court's limitation on § 102(b) not only ran afoul of *Baker v. Selden* but was also at variance with the unique facts of this case. Here, as the First Circuit explained, the words at issue are more fundamental to the operation of the program than even labels on buttons would be. 49 F.3d at 817, Pet. App. at 18a–19a. Here, the words of the menu command hierarchy are "essential to operating" the program and, hence, are part of the method of operation. *Id.* at 18a. As the First Circuit explained, "it would be impossible to operate [the Lotus program] without employing its menu command hierarchy." *Id.* at

19a. The holding of the First Circuit, narrowly tailored to the facts before it, is wholly consistent with similar cases in other circuits.

C. There Is No Conflict Among the Circuits That Menu Commands and Similar Methods of Operation Are Uncopyrightable.

Lotus argues that the First Circuit's reasoning is contrary to the law in other circuits, and that review of this case is needed to resolve a conflict in the circuits. To the contrary, there is no such conflict on the narrow issue actually presented here.

1. *The Ninth Circuit.* Lotus cites two cases for an alleged conflict between the First and Ninth Circuits. First, *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175-76 (9th Cir. 1989), is a code structure case and not pertinent here at all. Second, Lotus relies upon *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1477 (9th Cir.), *cert. denied sub. nom. BB Asset Mgmt. Inc. v. Symantec Corp.*, 113 S. Ct. 198 (1992), as allegedly creating a conflict. This is misplaced. That Ninth Circuit opinion specifically affirmed the lower court's holding that the menus at issue were "unprotectable under copyright." *Id.* at 1472. (While there is ambiguous *dicta* elsewhere in *Brown Bag* which Lotus cites for the proposition that "menus and keystrokes" are copyrightable, which the First Circuit also noted, *see* 49 F.3d at 819 n.14, Pet. App. at 22a n.14, that *dicta* does not alter the Ninth Circuit's ultimate opinion.)

In fact, a prior Ninth Circuit decision—not cited by Lotus—explicitly held that the menu commands of a spreadsheet software product were uncopyrightable. *Ashton-Tate Corp. v. Ross*, 916 F.2d 516 (9th Cir. 1990). In *Ross*, the plaintiff alleged that he had designed and given Ashton-Tate a complete menu hierarchy, including numerous submenus, which Ashton-Tate incorporated into its "Full Impact" spreadsheet product without compensating him. Unlike this case, Ross' list was handwritten, and had been developed before any computer code was written. However, like the Lotus 1-2-3 menu tree, Ross' tree contained both main menus and submenus. (A copy of Ross' actual menu

hierarchy was included in the district court record in this case. *See Borland II*, 799 F. Supp. at 220, Pet. App. at 136a.)

The *Ashton-Tate* district court ruled against Ross, finding that he was not entitled to compensation because the spreadsheet menu hierarchy was not entitled to copyright protection. Citing 17 U.S.C. § 102(b), the District Court held that Ross' "list of labels for user commands . . . is not protected under federal law." *Ashton-Tate Corp. v. Ross*, 728 F. Supp. 597, 602 (N.D. Cal. 1989). On appeal, Ross renewed his argument, the very same argument to justify copyrightability (*i.e* the presence of choices) advanced by the district court here. Ross argued that his menu command hierarchy evidenced

> numerous decisions by the authors about the ordering of the commands and their arrangement in the user interface. The fact that the authors of these design documents chose the order and groupings displayed, out of a nearly infinite number of possibilities, constitutes creative authorship.

Appellants' [Ross'] Opening Brief on Appeal at 25, 1st Cir. App. 1168. The Ninth Circuit confronted this argument directly and rejected it, stating that "[t]his argument is meritless for the reasons given in the district court's order, 728 F. Supp. at 602. The list simply does not qualify for copyright protection." 916 F.2d at 521-22.

Lotus does not mention *Ashton-Tate*, perhaps because the district court explicitly declined to follow it. *Borland II*, 799 F. Supp. at 220, Pet. App. at 136a. By reversing the district court and agreeing with *Ashton-Tate's* conclusion, the First Circuit removed any conflict with the Ninth Circuit, rather than creating one.

2. *The Tenth Circuit.* Lotus argues, and the First Circuit suggested, that the court's holding was in conflict with *dicta* in a footnote in *Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 n.23 (10th Cir.), *cert. denied*, 114 U.S. 307 (1993). A careful examination of *Autoskill* and subsequent Tenth Circuit authority reveals that there is no such conflict worthy of review at this time. Footnote 23 in *Autoskill* appears to hold that, for the purposes of a preliminary injunction, the district court did

not improperly enjoin a computer program where a student/user selected responses to the program's queries "by pressing the 1, 2, or 3 keys." 994 F.2d at 1495 n.23. The First Circuit noted this *dicta* and declined to "follow" the reasoning of this footnote. 49 F.3d at 813, 818–19, Pet. App. at 12a, 21a–22a.

To the extent that the *Autoskill* footnote bears on the issues of copyrightability of menus and was rejected by the First Circuit, it had *already been rejected* by the Tenth Circuit itself. In *Gates Rubber*, 9 F.3d 823, the Tenth Circuit limited its prior holding in *Autoskill* because that case only involved the review of a preliminary injunction order. *Id.* at 841. The Tenth Circuit vacated the *Gates Rubber* district court's finding that computer menus were copyrightable, and remanded the case to that district court for further consideration and analysis under the appropriate legal standards. *Id.* at 843–44.

The uncopyrightability of computer menus in the Tenth Circuit was confirmed by the very recent decision by the *Gates Rubber* district court, after the Tenth Circuit's remand. *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, No. 92-S-136 (D. Colo. filed June 12, 1995). Using the analysis ordered by the Tenth Circuit, the District of Colorado found that the menus of that program were uncopyrightable. Slip op. at 6–7. It saw no conflict among the circuits. To the contrary, the Colorado court approvingly cited the First Circuit's opinion in *Lotus v. Borland* without noting any conflict. *Id.* at 7. To the extent that *Gates Rubber* still poses any issues worthy of review, this Court can ultimately grant review of that case.[18]

---

[18] A few district courts have protected the menu aspects of screen displays by generally focusing on artistic and stylistic aspects of display, rather than the words themselves. For example, in *Digital Comm. Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 460 (N.D. Ga. 1987), the district court protected the "highlighting" and "capitalizing" of certain menus only because they "have no relationship to the functioning . . . of the computer program." Similarly, *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F. Supp. 984, 995-998 (D. Conn. 1989), provided protection to only the three screens that were not limited by functionality and denied copyright protection to the words and format of most menus. Where only the methods of operation were involved, district courts have declined to protect computer menus under the copyright laws. *See Mitek Holdings, Inc. v. Arce Eng'g Co.*, 864 F. Supp. 1568, 1579-80 (S.D. Fla. 1994).

3. *The Fifth Circuit.* Lotus also relies upon *Engineering Dynamics*, 26 F.3d 1335 (5th Cir. 1994), *supplemented on pet. for reh'g*, 46 F.3d 408 (5th Cir. 1995). Initially, *Engineering Dynamics* was the only court of appeals case which, like the district court opinions here, purported to apply an "abstractions"-like test *directly* to a "non-literal element." That particular non-literal element was the input formats to the computer program, and not a method of operating the program. Therefore, § 102(b) did not play any role in the court's decision. Indeed, the court did not base its reasoning on any analysis of § 102(b) or of methods of operation.

As discussed above, following the publication of the original opinion in *Engineering Dynamics*, the Fifth Circuit was deluged with requests for rehearing. It issued a supplemental opinion (not cited by Lotus) which greatly if not completely undercuts Lotus' arguments. The Fifth Circuit's supplemental opinion rejected the assertion that it protected the user formats in that case because there were "numerous ways the input formats could be organized." It instead stated that "[t]he panel did not say that in any case involving user interface the fact that the 'author' has selected from among possible formats is dispositive." 46 F.3d at 409. This is consistent with the First Circuit's views on one of the issues in dispute here, namely whether the availability of "expressive choices" in designing the menu command hierarchy makes the menus copyrightable. *See* 49 F.3d at 816, Pet. App. at 17a. Significantly, since the Fifth Circuit remanded that case for further proceedings, this Court can eventually review *Engineering Dynamics* if those proceedings result in any real conflict with the First Circuit.

4. *The Second and Third Circuits.* Finally, Lotus argues that this case conflicts with the Second and Third Circuit opinions in *Altai* and *Whelan*. As explained above, those cases involved the non-literal copying of code structure, rather than the "method of operation" issues involved here, and hence do not conflict with

the narrow issues presented by this case. Moreover, Lotus is incorrect that the First Circuit "rejected" *Altai's* abstraction-filtration-comparison test. The First Circuit did not reject *Altai*; to the contrary, it held that "the *Altai* test may provide a useful framework for assessing the alleged nonliteral copying of computer code." 49 F.3d at 815, Pet. App. at 14a. That issue is not present here, and could not properly be dealt with by this Court upon review. Finally, while *Altai* and numerous other courts have harshly criticized *Whelan*, *see* 982 F.2d at 705-06, that hardly makes *this* a suitable case to review the viability of *Whelan*. Indeed, since *Altai* is still pending before the Second Circuit, this Court can grant review of that case to address the *Altai/Whelan* debate.

II. THE FIRST CIRCUIT OPINION REMOVES THE UNCERTAINTY PRODUCED BY THE DISTRICT COURT AND UPHOLDS THE OVERALL INTELLECTUAL PROPERTY FRAMEWORK ESTABLISHED BY CONGRESS.

A. The First Circuit Opinion Restores Clarity and Predictability to the Law.

In the First Circuit, many of Borland's amici urged reversal of the district court because of the uncertainty created by the district court's methodology and result. It is ironic in the extreme that Lotus would now petition this Court, claiming it is the First Circuit's decision that has produced uncertainty. Software developers obviously need clear rules to enhance productivity. Prior to the district court's decisions, developers and their counsel, relying on § 102(b), *Baker v. Selden* and *Ashton-Tate v. Ross*, believed that menu commands and similar methods of operating a program were uncopyrightable. The First Circuit has now restored that clarity.

The alternative offered by Lotus, a case-by-case determination of whether § 102(b) means what it says, would have a chilling effect on software development. One need look no further than the facts of this case to understand the grave difficulties such a regime would portend. Here, two years, two opinions, hundreds of pages, and millions of dollars into this case, the district court

could still not determine the "precise scope of Borland's infringement," *Borland II*, 799 F. Supp. at 221, Pet. App. at 138a, without another year and one-half of proceedings. Under the regime proposed by Lotus, new and better products that compete for the business of a competitor's customers can be brought to market legally only at the cost of one's company. In rejecting such a regime, the First Circuit has restored clarity to at least a portion of the overall intellectual property protection framework established by Congress. Equally important is the fact that the First Circuit's opinion restores the long-standing demarcation between copyright and patent law.

Copyright is broad, long-lasting, easily obtainable protection. Copyright protection is obtained for copyrightable elements of a computer program merely by making a deposit of any copyrightable subject matter (such as code). *See* Copyright Office Circular 61, *Copyright Registration for Computer Programs*, at 2, 1st Cir. App. 1170. There is no examination procedure. The copyright lasts for approximately 75 years. Because copyright protection is so easy to obtain, and lasts so long, it was neither intended for, nor is it suited for, the granting of government-sanctioned monopolies for methods of operation. *See, e.g., Goldstein Treatise*, § 2.3.1 at 78, § 2.15.2 at 207; Paul Goldstein, *Infringement of Copyright in Computer Programs*, 47 U. Pitt. L. Rev. 1119, 1123-24 (1986). The monopoly on a menu command hierarchy or similar "method of operation" has far greater ramifications than even the monopoly on the "structure" of a program. Protecting code structure has no preclusive effect on the program's users. However, precluding a competitor's product from offering another's method of operation means that users will lose their investment in the skill set necessary to implement that method of operation if they switch to a competitor's product. 49 F.3d at 821, Pet. App. at 26a-27a.

Such a broad government-sanctioned monopoly must be secured, if at all, through the patent system. Patents on "methods of operation" are difficult to obtain and last a relatively short period of time (20 years or less). Patent applications must state the invention, describe the prior art, and set forth the claims for protection clearly and specifically. There is a complex examination process to ensure that the patentee will be contributing

something new to the state-of-the-art (*i.e.*, something novel and non-obvious, an advancement over the prior art, etc.), as the quid pro quo for the grant of monopoly. None of these safeguards are present in the copyright system established by Congress because it was (and is) not contemplated that the scope of copyright protection is tantamount to that of patent. Unless § 102(b) is recognized for what Congress intended it to be, the copyright law would afford over-extensive protection to works by applying only the most minimal level of scrutiny. The First Circuit's opinion is consistent with the intellectual property framework established by Congress and supports its rationale.

In short, Lotus lost this case because it tried to secure patent-type protection without satisfying the patent requirements of novelty, examination and contribution to the prior art. If, like the plaintiff in *Baker v. Selden*, Lotus cannot meet these requirements, or if it chooses not even to try, it should not be able to claim the same scope of protection through copyright law. That, as the *Baker* Court observed, "would be a surprise and a fraud upon the public. That is the province of letters-patent not of copyright." 101 U.S. at 102.

B. Both Copyright and Patent Remain Sources of Strong Protection, Providing Enormous Incentives for Authors and Inventors of Software Products to Innovate.

Lotus' suggestion that the First Circuit's ruling "could serve to roll back the scope of protection for computer programs generally," Pet. at 29, is wholly unfounded. Copyright protection remains a powerful and sufficient incentive for the development of new software. Copyright protects against piratical copying of object code. Copyright protects against appropriation of source code, either literally or by paraphrasing. Perhaps, in appropriate circumstances, copyright also protects against copying the detailed "structure" of another's program.

Nor does the First Circuit's opinion call into question the screen display portion of a "user interface." *Compare* Pet. at 28. As the *Altai* court observed, copyright protection for screen displays does not depend on the protection of "non-literal elements" of the computer program. Rather, copyright protects

against the unauthorized reproduction of "certain types of screen displays," that are "copyrighted separately as an audiovisual work." *Altai*, 982 F.2d at 703. The First Circuit did not even remotely suggest that screen displays are uncopyrightable. Rather, its holding that menu commands are uncopyrightable does not interfere with the protection of screen displays "because the way the screens look has little bearing on how users control the program." 49 F.3d at 816 & n.10, Pet. App. at 16a & n.10.

Menu commands and similar methods of operating the program continue to be protected by the patent law, as they always have been. The record below contains several examples of menu command hierarchies, including those of IBM (Lotus' new owner) that are protected by utility patents. Borland S.J. Brief, Exs. 16 & 23 (U.S. Patents Nos. 4,989,141, 4,611,306). Indeed, the U.S. Patent and Trademark Office has recently announced it is changing the rules for patentability of software, making it even easier to obtain software patents.[19]

CONCLUSION

This Court has made it clear that only Congress may redraw the balance between private monopoly and public access. The courts are required to defer to Congress "when major technological innovations alter the market for copyrighted materials." *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 431 (1984). If Congress has not expressly chosen to expand the scope of copyright protection, it is not the job of the courts to do so. On the contrary, "[i]n a case like this, in which Congress has not plainly marked our course, we must be circumspect in construing the scope of rights created by a legislative enactment which never contemplated such a calculus of interests." *Id.* The district court

---

[19] *See* U.S. Department of Commerce, Press Release No. 95-21, *Software Patent Guidelines Released Today*, 6/1/95, 60 Fed. Reg. 28,778 (proposed June 2, 1995); B. Rosewicz, *Patent Office Acts to Clarify Software Rules*, Wall St. J., June 2, 1995, at A1; U.S. Department of Commerce, Press Release No. 95-18, *USPTO to Develop Guidelines to Protect Software Inventions*, 3/30/95; M. Betts, *Feds to ease software patent guidelines*, Computerworld, April 17, 1995, at 20.

in this case plainly thought that it was empowered to create new law in "uncharted" territory and to "draw the line between copyrightable and non-copyrightable elements of computer programs." *Borland I*, 788 F.Supp. at 90, Pet.App. at 165a; *Paperback*, 740 F.Supp. at 53, Pet.App. at 206a. In so doing, the district court usurped the role of Congress. The First Circuit corrected this error.

Twenty years ago, the first personal computers had no screens or keyboards; the users operated the machines by pressing buttons or switches on the front of the machines. No one would ever claim that such buttons were copyrightable. Twenty years from now, users will operate personal computers with spoken words, and without any physical buttons or keyboards. It is inconceivable that anyone could claim that such spoken methods of operation will be copyrightable. At the intermediate stage of technology relevant here, Lotus used typed words as the buttons or switches to operate its spreadsheet program. Those words are no more copyrightable than physical buttons were twenty years ago, or than spoken commands will be twenty years from now.

For the foregoing reasons, the petition for a writ of certiorari should be denied.

Respectfully submitted,

GARY L. REBACK
(Counsel of Record)
MICHAEL BARCLAY
ISABELLA E. FU
WILSON, SONSINI,
GOODRICH & ROSATI
Professional Corporation
650 Page Mill Road
Palo Alto, California 94304
Telephone: (415) 493-9300

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY &
GELHAAR
One Post Office Square
33rd Floor
Boston, Massachusetts 02109
Telephone: (617) 542-7400

Attorneys for Respondent

July 6, 1995

(6)

No. 94-2003

Supreme Court, U.S.
FILED
JUL 18 1995
CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

PETITIONER'S REPLY IN SUPPORT OF PETITION FOR A WRIT OF CERTIORARI

CHARLES FRIED
1575 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4636

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

Counsel for Petitioner

BEST AVAILABLE COPY

19pp

AMENDED RULE 29.1 STATEMENT

On July 5, 1995, Lotus Development Corporation became a wholly owned subsidiary of White Acquisition Corporation, itself a wholly owned subsidiary of the International Business Machines Corporation ("IBM").

IBM has no parent corporation and no subsidiaries that are not wholly owned, except for certain foreign subsidiaries in which a minimal amount of shares (fewer than 1%), which are not publicly traded, are held by foreign nationals in accordance with local laws, except for IBM World Trade Corporation, a wholly owned subsidiary of IBM, which owns, directly or indirectly, more than 50 percent but less than 100 percent of the following corporations:

ABMCG Architecture and Business Models
Consulting Group S.p.A.
AD/Consultants
Agritalia S.r.l.
APEM S.A.
A/S Datalog F.P.
AUREA
AXONE S.A.
BLU A.G.E.S. Puglia e Basilicata S.r.l.
Boselli Sistemi NBS S.p.A.
Bridgeware Europe B.V.
Central Services Group B.V.
CGI Consulting, S.A.
CGI Consulting S.p.A.
CGI Corporacion S.A.
CGI Corporation Inc.
CGI Corporation Limited
CGI General de Industria, S.A.
CGI Informatica S.A.
CGI Informatica Holding BV
CGI Informatica S.r.l.
CGI Informatik GmbH

CGI Informatique S.A.
CGI Interprogram GmbH
CGI Limited
CGI LS2 Limited
CGI Portugal S.A.
CGI Services
CGI Systems B.V.
CGI Systems Inc.
CGI Systems Plc
CGI Systems S.A.
CIMAD Consultants N.V.
City Info Finland Oy
CR2A Holding
CR2A-DI
CR2A Selection
CS Technical Staff Co., Ltd.
DST Logistica S.r.l.
Datablue
Didao
EDB Gruppen A/S
EDB Gruppen Herning
EDB Gruppen K benhavn Holding A/S
EDB Gruppen Oslo
EDB Gruppen Silkeborg A/S
Ediat
EG af 25.10.91 ApS
EG Data Inform A/S
EG Data Inform Prod. A/S
EG Gruppen Kolding A/S
EG Image A/S
Electronic Business Machines (EBM)
Eurequip S.A.
FINEST S.A.
GMV Conseil
HG Gruppen Hjorring A/S
I-LINIE Software GmbH

IBD Informationssysteme Betriebs-
und Dienstleistungs-GmbH
IBM Japan GB Kansai Solution Co., Ltd.
IBM Japan GB Solution Co., Ltd.
IBM Japan GB Tokai Solution Co., Ltd.
IBM South Africa Group Ltd.
IBM South Africa (Pty) Ltd.
IBM Unternehmensberatung GmbH
ICG Informationssysteme Consulting und
Betriebs-Gesellshaft GmbH
ID Centeret Hj.
ID Centeret Hjorring
IDG Informationsverarbeitung und
Dienstleistungen GmbH
Industria Per IL Software S.p.A.
Integrated Systems Solutions Corporation Australia Ltd.
International Information Products
(Shenshen) Company Ltd.
International Outsourcing Services (Pty) Ltd. (IOS)
IPN Informatica per Il Negozio S.r.l.
ISD Informationssysteme und Dienstleistungs GmbH
IT Partner Herning A/S
Japan Business Logistics Co., Ltd.
Logic Line Operations S.A.
Logiciels Sirc Inc.
Logon Solution Integrator (Logon S.I.)
MAS A/S
MAS Worldwide Limited
MONTICS, S.A.
NSI S.r.l. Nuove Soluzioni Informatiche S.r.l.
NULOGIX Technical Services Inc./Services
Techniques NULOGIX Inc.
PARTITIEL S.A.
Personal Computer Co., Brasil S.A.
Polar Bear Software Corporation
Prodstar Benelux BV
Prodstar (Deutschland) GmbH

Prodstar Espana S.A.
Prodstar GB Limited
Prodstar Portugal Lda.
Prodstar Production Software
Prodstar S.A.
Progipart, S.A.
PYRENEES INFORMATIQUE S.A.
R. J. Thompson Data Systems Inc.
Rigorsoft Solucoes Informaticas, S.A.
Service Maintenance Systems (SMS) S.A.
Service Plus Ltd.
SETIN S.r.l.
SISTEMI INFORMATIVI S.p.A.
Siven S.r.l.
Skillteam S.A.
Societe Civile Immobiliere Monge
Softcycle S.A.
Sulzer Informatik AG
Tiltan System Engineering Ltd.
Tunisian Business Machines (TBM)
Viamet B.V.
V.T. Nielsen Data A/S

TABLE OF CONTENTS

	PAGE
AMENDED RULE 29.1 STATEMENT	i
TABLE OF AUTHORITIES	vi
ARGUMENT	1
CONCLUSION	10

TABLE OF AUTHORITIES

Cases PAGE

Ashton-Tate Corp. v. Ross, 916 F.2d 516 (9th Cir. 1990) 7, 7n, 8

Ashton-Tate Corp. v. Ross, 728 F. Supp. 597 (N.D. Cal. 1989) 7-8, 8n

Autoskill, Inc. v. National Educational Support Systems, Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 U.S. 307 (1993) 8

Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), *cert. denied sub nom. BB Asset Mgmt. Inc. v. Symantec Corp.*, ___ U.S. ___, 113 S. Ct. 198 (1992) 6, 7, 10n

Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 9, 9n

Engineering Dynamics Inc. v. Structural Software Inc., 26 F.3d 1335 (5th Cir. 1994), *pet. for reh'g en banc denied*, 46 F.3d 408 (5th Cir. 1995) 5, 6, 10n

Feist Publications, Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991) 5

Gates Rubber Co. v. Bando Chemical Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) 8, 10n

Gates Rubber Co. v. Bando Chemical Indus., Ltd., No. 92-S-136 (D. Colo. filed June 12, 1995) 8-9

Johnson Controls, Inc. v. Phoenix Control Systems, Inc., 886 F.2d 1173 (9th Cir. 1989) 6

Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990) 8

PAGE

Statutes

17 U.S.C. § 101 ..... 3, 3n, 4

17 U.S.C. § 102 ..... 1, 4, 9

Other Authorities

Anthony L. Clapes, Patrick Lynch and Mark R. Steinberg, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. Rev. 1493 (1987) 1n

Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright Over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559 (1994) 10n

Arthur R. Miller, *Copyright Protection for Computer Programs, Databases and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 997 (1993) 10n

Robert D. Sprague, *Multimedia: The Convergence of New Technologies and Traditional Copyright Issues*, 71 Den. U.L. Rev. 635 (1994) 10n

Nicholas P. Terry, *GUI Wars: The Windows Litigation and the Continuing Decline of "Look and Feel"*, 47 Ark. L. Rev. 93 (1994) 10n

Borland's Brief fails to confront the central issue presented by this case: whether, and to what extent, the Congressional mandate extending copyright protection to computer programs protects expressive elements of user interfaces, as well as programming code. It then distorts the pertinent case law in an effort to show that the First Circuit's unprecedented use of Section 102(b) to deny protection to such expressive elements does not conflict with the law of other circuits. Borland concedes that a uniform, national answer to these questions is of vital importance to the software industry (Borland Br. at 24), but it fails to demonstrate that the First Circuit has provided any usable answer, much less one that has nationwide acceptance.

1. Borland's position rests upon three false premises.

First, Borland repeatedly mischaracterizes the nature of Lotus' claims and the scope of the district court's holdings. Lotus did not claim, and the district court did not rule, that Lotus' copyrights gave it a monopoly on the use of any individual menu command such as "COPY," "MOVE," or "PRINT." *See* Borland Br. at 1, 5 & 8. Rather, the issue at all times has been the copyrightability of the Lotus 1-2-3 menu tree or command hierarchy *taken as a whole*—a collection of 469 different menu commands, organized and sequentially arranged into more than 50 different menus. This distinction is critical. It is, in fact, the structure and arrangement of the *entire* 1-2-3 menu hierarchy that Borland copied in its products and that the district court found to contain sufficient original expression to be copyrightable. Pet. App. at 86a-89a. Borland's focus upon individual commands is a classic example of what Professor Nimmer called the fallacy of *reductio ad absurdum*: if the first word of a work is not protectable, standing alone, and the second is not, and so on, then the entire work must be unprotected.[1] Yet, as is true for any literary work, it is the com-

---

[1] Declaration of Melville B. Nimmer, dated November 15, 1984, at ¶ 14, appended to Anthony L. Clapes, Patrick Lynch and Mark R. Steinberg, *Silicon Epics and Binary Bards: Determining the Proper*

bination and arrangement of the individual words in the 1-2-3 menus as a whole that give them their context, meaning, and communicative power—and that make them copyrightable.

Second, Borland misdescribes the Lotus 1-2-3 user interface in an attempt to reinforce the First Circuit's erroneous analogy comparing the words displayed in the 1-2-3 menus to the buttons on a machine. *See* Pet. App. at 18a-19a. It is simply untrue that a 1-2-3 user can instruct the program to perform an operation by typing out a menu command, such as "PRINT," as Borland represents. Borland Br. at 8. Neither are the words "used as buttons to operate a program." *Id.* at 13. Rather, the words serve to explain to users what their available command choices are at a given point in the operation of the program, and to identify the keystrokes that will communicate those choices to the program.[2] It is the keystrokes— not the words appearing on the screen—that cause the program to take any action. Indeed, Lotus could have designed a user interface that did *not* display such information to the user (Pet. App. at 286a-288a); users could have been forced, for example, to remember the permissible keystrokes and to rely upon printed documentation for any necessary reminders. That this table of contents conveniently appears on the screen instead should not transform the character of those words from their fundamentally informative purpose into the equivalent of a mechanical device.

Recognition of this distinction also provides the appropriate perspective from which to consider Borland's argument that the 1-2-3 menu command hierarchy should be protected, if at all, under patent rather than copyright law. Borland Br. at 25-26. According to Borland, copyright in a computer program covers only its code, and "perhaps" the "detailed 'structure' " of its

Scope of Copyright Protection for Computer Programs, 34 UCLA L. Rev. 1493, 1585, 1588-89 (1987).

[2] The permissible keystrokes include the first letters of the words appearing in a particular menu or the "ENTER" key, which will communicate the currently highlighted menu command.

code. *Id.* at 26. That is, in Borland's view, copyright properly protects the digital notations that operate within the machine causing it to function, but *not* the English words that appear on the screen to assist a person in using the program, because the latter (but not the former) is a "method of operation." *Id.* at 18-19. Thus, according to Borland, an arrangement of words conveying information to human beings must be protected by patent, while machine-level communication is covered by copyright. Far from restoring "the overall intellectual property protection framework established by Congress," as Borland claims (*id.* at 25), this proposal would turn it on its head.

Third, Borland depicts the 1-2-3 menu command hierarchy as distinct from the computer program that generates it, drawing a bright line between a "computer program" (which it defines as consisting only of programming code) and any informational content that the same code is designed to display. *Id.* at 17.[3] Borland's entire legal analysis rests upon this distinction, which it purports to draw from the definition of a "computer program" found in Section 101 of the Copyright Act. 17 U.S.C. § 101. That definition—"a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result"—contains, on its face, no limitation that restricts its applicability to programming code. The 1-2-3 menu commands fit the statutory definition equally well, as they express a set of instructions to be used in a computer to produce particular results.[4] Certainly from the perspective of a typical user any suggestion that the user inter-

[3] Contrary to Borland's suggestion (*id.*), the facts of this case prove this to be incorrect. The district court received in evidence the print-out of a particular file, stored internally in the Borland programs, in which every word from the 1-2-3 menus, arranged as in the 1-2-3 menu hierarchy, was spelled out. *See* Pet. App. at 34a-35a.

[4] The menu commands also fit the literal definition of a "literary work" found in the same section: "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." 17 U.S.C. § 101.

face is not an integral part of the computer program seems absurd.[5]

2. Borland's argument also reveals a basic misperception of the role of the Section 101 definitions in the overall scheme of the Copyright Act. The definitions neither delimit the bounds of copyrightable subject matter, nor specify the scope of protection afforded to copyrightable works. That is the purpose of Section 102(a), with its illustrative but not limitative list of the types of works eligible for copyright, and of the limitations found in Section 102(b). It is no more accurate to employ the definition of a "computer program" in Section 101 as a basis for precluding protection for menus that a program displays, than to use the statutory definition of a "motion picture"—defined as a "series of related images which, when shown in succession, impart an impression of motion" (17 U.S.C. § 101)—to preclude protection for a motion picture's plot, dialogue, or characters, all of which plainly are copyrightable and covered by the copyright in the motion picture itself.

Borland agrees that Section 102(b) is "the legislative *embodiment* of the idea/expression dichotomy," which supplies the "'line' between copyrightable and uncopyrightable subject matter." Borland Br. at 18 (emphasis in original). But the First Circuit disregarded the district court's finding that the 1-2-3 menu hierarchy contains expression separable from its underlying idea. Pet. App. at 17a. Beyond its misplaced reliance on the definition of a "computer program" in Section 101, Borland proffers no reason why the 1-2-3 menu command hierarchy can

[5] Neither does the Copyright Office share Borland's view on this subject. In its *amicus* brief, the Copyright Office declared: "Menu screens are integral parts of computer programs, and they may or may not contain copyrightable expression." Brief Amicus Curiae on Behalf of the Register of Copyrights, at 5. Contrary to Borland's contention (Borland Br. at 4), this brief was *not* filed in the court of appeals, urging reversal, but in the district court, and in it the Copyright Office expressly declared that its "brief should not be taken as commenting on the copyrightability or scope of protection to be accorded the works at issue in this case, i.e., those relating to Lotus 1-2-3." *Id.* at 3.

be considered an unprotected "method of operation" that would not, under the First Circuit's reasoning, also preclude copyright protection for computer programs generally. The uncertainty confronting the software industry in the wake of the First Circuit's decision is both genuine and profound,[6] and Borland's Brief does nothing to obviate the need for this Court's resolution of the issue.

3. Borland misreads the pertinent decisions from other circuits to argue that the First Circuit's decision presents no conflict.

a. Borland inaccurately portrays the Fifth Circuit's denial of a petition for rehearing en banc in *Engineering Dynamics Inc. v. Structural Software Inc.*, 26 F.3d 1335 (5th Cir. 1994), *pet. for reh'g en banc denied*, 46 F.3d 408 (5th Cir. 1995), as retreating from its earlier opinion extending copyright protection to a computer program's user interface. Borland Br. at 23. In seeking rehearing of the Fifth Circuit's decision, the defendant Structural Software and its amici argued—as do Borland and many of the same amici in this case—that the court erroneously had based its original decision solely upon the existence of alternatives to plaintiff's choice of input formats. The Fifth Circuit rejected that characterization of its rationale as "overly simplistic" (46 F.3d at 409), observing that its conclusion rested upon many facts, including a finding that the plaintiff's user interface reflected a sufficient degree of creativity to qualify as original expression under this Court's ruling in *Feist Publications, Inc. v. Rural Telephone Serv. Co.*, 499 U.S. 340, 358 (1991). *Engineering Dynamics*, 46 F.3d at 409. The court neither modified nor withdrew any of its previous rulings on copyrightability, including its reliance on Judge Keeton's decisions below. To the contrary, the Fifth Circuit reaffirmed its direction to the district court to apply the abstrac-

[6] *See* Brief Amicus Curiae of Intellectual Property Owners in Support of Petitioner, at 9-10; Brief Amicus Curiae of Information Technology Industry Council in Support of the Petition for a Writ of Certiorari, at 3 & 10-11.

tion-filtration-comparison test to determine the precise scope of protection in plaintiff's user interface, in particular by identifying elements that were functionally dictated. *Id.* at 410.

In this case, Judge Keeton specifically found, after trial, that the 1-2-3 menu command hierarchy was *not* dictated by the functionality it allowed users to achieve, in addition to finding that numerous viable alternatives existed to that particular menu command hierarchy. Pet. App. at 89a, 93a & 130a. Under the Fifth Circuit's ruling in *Engineering Dynamics*, these findings would compel a conclusion of copyrightability. The First Circuit, in contrast, either ignored these findings or deemed them irrelevant to its analysis. *Id.* at 17a. The conflict is square. Indeed, Borland argued before the First Circuit that the *Engineering Dynamics* court erred when it "adopted" Judge Keeton's reasoning and held that a computer program's user interface *is* copyrightable.[7]

b. Borland fares no better in its attempt to depict the law of the Ninth Circuit as not in conflict with the First. Borland casually dismisses the Ninth Circuit's decision in *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173 (9th Cir. 1989), as "not pertinent here at all" (Borland Br. at 20), despite the court's explicit statement that:

> [w]hether the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea or an idea itself. (*Id.* at 1175.)

Borland similarly mischaracterizes the Ninth Circuit's decision in *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465 (9th Cir.), *cert. denied sub nom. BB Asset Mgmt. Inc. v. Symantec Corp.*, ___ U.S. ___, 113 S. Ct. 198 (1992). The district court did not hold "that the menus at issue were 'unprotectable by

[7] Letter of Gary L. Reback, counsel of record for Borland, to Clerk of the Court for the First Circuit, dated September 27, 1994.

copyright' " (Boland Br. at 20), and neither did the Ninth Circuit. Borland refers to a portion of the district court's decision, as summarized by the Ninth Circuit, rejecting plaintiff's claim that its copyright protected certain functional "features" of the program, such as the ability to access files or to print. 960 F.2d at 1472.[8] The district court carefully distinguished these unprotectable "features" from the "expression" in "the programs' respective opening menus," which it found were "not substantially similar". *Id.* at 1473. The Ninth Circuit recognized this distinction, affirming the district court's test for comparing "the expressions embodied" in the programs' menus. *Id.* at 1475. That the Ninth Circuit considered the menus to comprise protected expression is obvious throughout its analysis. They simply had not been copied in that case.

Borland faults Lotus for not discerning a different rule in the Ninth Circuit's earlier decision in *Ashton-Tate Corp. v. Ross*, 916 F.2d 516 (9th Cir. 1990). Lotus is not alone in concluding that this decision has no bearing upon the issues in this case. The First Circuit did not mention that case, and the Ninth Circuit omitted any such reference when it discussed the menus at issue in *Brown Bag* two years later. Contrary to Borland's assertion, the Ninth Circuit did *not* hold in *Ross* that "the menu commands of a spreadsheet software product were uncopyrightable" (Borland Br. at 20); that question was not before the court. Rather, the issue was whether Ross, having furnished a document consisting of "only a list of labels for user commands" that he "thought should be included" in a predecessor to the Ashton-Tate program,[9] could claim to be a joint author of the program's eventual user interface. *Ashton-*

[8] In this case, Judge Keeton drew the same distinction, specifically finding that 1-2-3's "features" or functional capabilities comprise part of the unprotected "idea" underlying the 1-2-3 menu command hierarchy, rather than part of the protected expression. Pet. App. at 129a.

[9] Judge Keeton explicitly considered this and other facts in the course of distinguishing this case from *Ross* in his analysis (Pet. App. at 135a-36a), and did not disregard the *Ross* decision merely because it arose in another circuit, as Borland suggests. Borland Br. at 6.

Tate Corp. v. Ross, 728 F. Supp. 597, 602 (N.D. Cal. 1989). The Ninth Circuit's holding was that this handwritten list—*not* the program's menu commands and hierarchy—did not qualify for copyright protection, and thus Ross had no claim to joint authorship in the program. 916 F.2d at 522.[10]

c. Borland further errs by asserting that the Tenth Circuit "rejected" its decision in *Autoskill, Inc. v. National Educational Support Systems, Inc.*, 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 U.S. 307 (1993), five months later in *Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, 9 F.3d 823 (10th Cir. 1993). Borland Br. at 22. In *Gates Rubber*, the Tenth Circuit explicitly adopted the abstraction-filtration-comparison test "which we previously approved for use in the context of a preliminary injunction ruling in *Autoskill*," also citing with approval Judge Keeton's decision in *Lotus Development Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990). *Gates Rubber*, 9 F.3d at 834. The court of appeals directed the district court to apply that test to determine the copyrightability of plaintiff's menus (*id.* at 843-44)—a senseless direction if the Tenth Circuit agreed with the First Circuit that menus are *per se* ineligible for copyright. Thus, under the law of the Tenth Circuit but not the First, menus may be protected to the extent they contain identifiably separable expression. This rule is wholly unaffected by the district court's subsequent factual finding that the menus at issue in *Gates Rubber* fell "towards the idea end of the idea-expression dichotomy" because the "universe of choices to accomplish [their] task is relatively small." *Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, No. 92-S-136, slip.

[10] Among the reasons given by the district court for its conclusion was the fact that Ross "did not contribute in any manner" to the "implementation of his suggested commands" in the program, and therefore had contributed only an "idea" rather than expression to the creation of the program's user interface. 728 F. Supp. at 602. Borland glides over this and other critical facts necessary to a proper understanding of the *Ross* decision by use of ellipsis (Borland Br. at 21), which omits, between the first six and last five words of the quote, five complete sentences and virtually all of the district court's explanation for its conclusion.

op. at 7 (D. Colo. filed June 12, 1995). In contrast, Judge Keeton in this case found that the possible "universe of choices" for a spreadsheet menu hierarchy providing the same functionality as in Lotus 1-2-3 was virtually unlimited (Pet. App. at 131a), although, as noted above, his decision did *not* rest upon that finding alone. *Id.* at 93a, 129a, 130a & 133a.

d. Borland also claims that the Second Circuit "pointedly *rejected*" Judge Keeton's *Paperback* analysis in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992). Borland Br. at 4 (emphasis in original). In fact, the Second Circuit cited *Paperback* with approval in support of critical steps in its own reasoning no fewer than five times.[11] Borland ignores these substantive citations in favor of the court's single reference in *dicta* concerning the potential policy implications of the Second Circuit's decision, leading to a conclusion that courts must apply, "in accordance with Congressional intent, long-standing principles of copyright law to computer programs." 982 F.2d at 712. It is precisely these "long-standing principles"—including the principle that Section 102(b) must be interpreted by applying the idea/expression dichotomy to *each* element of a copyrighted work for which protection is sought—that both the Second Circuit and the district court below espoused and followed, but the First Circuit ignored.

Borland accuses Judge Keeton of having rejected or ignored well-settled principles in order to achieve a "controversial extension of copyright law" in this case. Borland Br. at 3, 11.

[11] Specifically, the Second Circuit cited *Paperback* for its exposition of the pertinent legislative history (982 F.2d at 702); for the principle that non-literal elements of computer programs, like other literary works, are protected by copyright (*id.* at 703); for the conclusion that the idea/expression distinction should be used to determine which aspects of computer programs are copyrightable (*id.* at 704); for its use of merger doctrine to address the question of substantial similarity "in the context of computer program structure" (*id.* at 709); and, explicitly agreeing with Judge Keeton's "approach," for the conclusion that merger is "an effective way to eliminate non-protectable expression contained in computer programs." (*Id.*)

This is not correct. Faithful to the Congressional mandate, Judge Keeton applied traditional copyright doctrine to the task of defining the scope of protection available to computer programs—a new, high technology form of "literary work". He took as his guide the landmark decisions of Judge Learned Hand that have, for decades, instructed the courts in the proper use of the idea/expression dichotomy, the merger doctrine and other time-tested analytic tools to distinguish the protected from the unprotected elements of literary works. The result was an analysis that not only has influenced the tests subsequently adopted in decisions of the Second, Fifth, Ninth, and Tenth Circuits, as those courts have all acknowledged, but also has generated considerable favorable commentary in the academic community.[12] Judge Keeton's conclusion that a computer program's menu command hierarchy is copyrightable where, as here, its original expression is separable from its underlying functionality, is consistent with decisions from across the land.[13] The First Circuit is the first court to hold otherwise.

CONCLUSION

For these reasons and for the reasons set forth in the petition, the petition for a writ of certiorari should be granted.

---

12 *See, e.g.*, Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright Over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559, 2571 n. 68 (1994); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 997 (1993); Robert D. Sprague, *Multimedia: The Convergence of New Technologies and Traditional Copyright Issues*, 71 Den. U.L. Rev. 635 (1994); Nicholas P. Terry, *GUI Wars: The Windows Litigation and the Continuing Decline of "Look and Feel"*, 47 Ark. L. Rev. 93 (1994).

13 *See, e.g., Engineering Dynamics*, 26 F.3d at 1343-44; *Gates Rubber*, 9 F.3d at 849; *Brown Bag*, 960 F.2d at 1475-76.

Respectfully submitted,

CHARLES FRIED
1575 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4636

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

MOTION FILED
JUN 6 1995

No. 94-2003

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE AND BRIEF AMICUS CURIAE OF INTELLECTUAL PROPERTY OWNERS IN SUPPORT OF PETITIONER

MORTON DAVID GOLDBERG
GARY L. GRISWOLD
HERBERT C. WAMSLEY *
INTELLECTUAL PROPERTY OWNERS
1255 Twenty-Third Street, N.W.
Suite 850
Washington, D.C. 20037
(202) 466-2396
Attorneys for Amicus Curiae
* Counsel of Record

WILSON - EPES PRINTING CO., INC. - 789-0096 - WASHINGTON, D.C. 20001

BEST AVAILABLE COPY

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE

Intellectual Property Owners ("IPO"), a not-for-profit national organization founded in 1972 to represent the interests of owners of intellectual property in the United States, respectfully moves for leave to file the attached brief *amicus curiae* in this copyright infringement case. The consent of the attorney for petitioner has been obtained. The consent of the attorney for respondent was requested but refused.

Membership in IPO currently includes nearly one hundred large and mid-size companies and more than three hundred small businesses, universities, independent inventors, authors, executives and attorneys who own or are interested in patents, trademarks, copyrights and other intellectual property rights. Many of IPO's members develop computer software, either to compete in the worldwide software market, to complement the services they provide their customers, or to support other segments of their businesses.

Computer software development has become a key part of the United States economy. As with all copyright industries, it relies on appropriate levels of copyright protection for the works it creates. Because the market for computer programs is nationwide, the industry also relies on a measure of national uniformity in the application of copyright law. The First Circuit's decision below contravenes both of these important industry expectations.

The First Circuit has denied copyright protection to expression in one of the most creative elements of a computer program—the interface between the computer and the user—because that element performs a useful function. Even though copyright protection was conferred on authors of computer programs by Congress, the First Circuit's decision threatens to deny that protection to the expression of any program element that also performs a useful function.

The First Circuit's interpretation of Section 102(b) of the Copyright Act, on which it based its denial of copyright protection to the Lotus 1-2-3 user interface, conflicts with the interpretation of that provision in other circuits. This conflict subjects the works of computer program authors to flatly contradictory legal rules within a national marketplace.

This case is much more than a dispute between the parties: it is a clash between two fundamentally-opposed views of what protection the Copyright Act provides for computer software. That is why this case has attracted such extraordinary attention within the computer industry —as the ten briefs *amicus curiae* filed in the First Circuit attest.

One view embraces the Congress' decision that computer programs are literary works, to be afforded protection under copyright law in accordance with the same principles that apply to all copyrighted works. In this view, appropriate protection under copyright law fosters innovative competition in the software industry and fulfills the Constitutional dictate "to promote the progress of science and useful arts." U.S. Const. art. I, § 8, cl. 8. IPO holds this view, together with the greater part of the software industry.

The opposite view rejects Congress' decision that programs should be protected under copyright. By attempting to erect special judicial barriers to protection of computer programs, those expressing this view seek to curtail the protection that the copyright statute confers. They do so to foster imitative competition—competition founded on copying the fruits of others' innovative development.

As an association representing some of the leading innovators in the software industry, IPO is in a unique position to bring to the Court's attention the viewpoint of the major portion of the business sector that is most affected by the decision below. IPO is in a better position than the parties to assist the Court in assessing the important legal and policy implications of this case for the software industry and the copyright system. IPO's brief *amicus curiae* focuses on these considerations. IPO respectfully requests that its motion be granted and that its brief be considered by the Court in connection with the petitioner's request for the grant of a Writ of Certiorari.

Respectfully submitted,

MORTON DAVID GOLDBERG
GARY L. GRISWOLD
HERBERT C. WAMSLEY *
INTELLECTUAL PROPERTY OWNERS
1255 Twenty-Third Street, N.W.
Suite 850
Washington, D.C. 20037
(202) 466-2396
Attorneys for Amicus Curiae

July 7, 1995

* Counsel of Record

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	iii
INTEREST OF AMICUS CURIAE	1
ARGUMENT	3
I. THE DECISION BELOW CONFLICTS WITH DECISIONS OF OTHER CIRCUITS AND CREATES UNCERTAINTY AS TO COPYRIGHT PROTECTION OF COMPUTER PROGRAMS	5
A. The First Circuit's Holding on Copyrightability of User Interfaces Conflicts With Decisions of Three Other Circuits	5
B. Applied to Computer Programs, the First Circuit's Interpretation of 17 U.S.C. Section 102(b) Conflicts With Decisions of at Least Five Other Circuits	7
C. A Conflict in Rules of Copyrightability Leaves Program Authors Uncertain of Their Rights and Discourages Creation of Computer Programs	9
II. THE DECISION BELOW DENIES COMPUTER PROGRAMS THE COPYRIGHT PROTECTION MANDATED BY CONGRESS AND DIMINISHES THE STATUTORY INCENTIVES TO CREATE THEM	10
A. The First Circuit Misinterpreted a Key Provision of the Copyright Act Because It Failed to Employ Appropriate Means of Statutory Construction	11
B. Denying Protection to Expression in Computer Programs Erodes Severely the Incentives Congress Enacted for Their Authors	17

TABLE OF CONTENTS—Continued

Page

C. The First Circuit's Construction of Section 102(b) Is Such a Significant Departure From That of the Overwhelming Majority of Courts of Appeals and District Courts as to Call for an Exercise of This Court's Power of Supervision 19

CONCLUSION 20

TABLE OF AUTHORITIES

Cases Page

Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984) 18

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) 7

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, —— U.S. ——, 115 S. Ct. 1176 (1995) 6, 17

Atari Games Corp. v. Nintendo of America, Inc., 975 F.2d 832 (Fed. Cir. 1992) 8

Atari Games Corp. v. Oman, 888 F.2d 878 (D.C. Cir. 1989) 5

Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982) 17

Autoskill, Inc. v. National Educational Support Systems, Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, —— U.S. ——, 114 S. Ct. 307 (1993) 5, 17

Baker v. Selden, 101 U.S. 99 (1879) 16

Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), *cert. denied*, —— U.S. ——, 113 S. Ct. 198 (1992) 6

Cable/Home Communication Corp. v. Network Products, Inc., 902 F.2d 829 (11th Cir. 1990) 17

CCC Information Services, Inc. v. Maclean Hunter Market Reports, Inc., 44 F.3d 61 (2d Cir. 1994), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995) 17

Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 8

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), *opinion supplemented on denial of rehearing*, 46 F.3d 408 (5th Cir. 1995) 6, 8

FDIC v. Meyer, 510 U.S. ——, 114 S. Ct. 996 (1994) 12

Florida v. Rodriguez, 469 U.S. 1 (1984) 20

TABLE OF AUTHORITIES—Continued

Page

Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823, 834 (10th Cir. 1993) 8
Gustafson v. Alloyd Co., —— U.S. ——, 115 S. Ct. 1061 (1995) 13
Harper & Row, Publishers v. Nation Enterprises, 471 U.S. 539 (1985) 19
Kelly v. Robinson, 479 U.S. 36 (1986) 11
Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527 (5th Cir.), *cert. denied*, —— U.S. ——, 115 S. Ct. 82 (1994) 17
Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807 (1st Cir.), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995) (No. 94-2003) ..... *passim*
Lotus Development Corp. v. Borland International, Inc., 788 F. Supp. 78 (D. Mass. 1992), *rev'd*, 49 F.3d 807 (1st Cir.), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995) (No. 94-2003) 6
Lotus Development Corp. v. Paperback Software, International, 740 F. Supp. 37 (D. Mass. 1990) .. 6
M. Kramer Manufacturing Co. v. Andrews, 783 F.2d 421 (4th Cir. 1986) 7, 17
Mazer v. Stein, 347 U.S. 201 (1954) 18
Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931) 8
Robert R. Jones Associates, Inc. v. Nino Homes, 858 F.2d 274 (6th Cir. 1988) 17
Toro Co. v. R & R Products Co., 787 F.2d 1208 (8th Cir. 1986) 17
Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 17

Constitutional Provision, Statutes and Public Laws

United States Constitution, art. I, § 8, cl. 8 18
17 U.S.C. § 101 ..... 4, 12, 15

TABLE OF AUTHORITIES—Continued

Page

17 U.S.C. § 102(a) 12, 13, 17
17 U.S.C. § 102(b) ..... *passim*
Pub. L. No. 93-573 (1974) 13
Pub. L. No. 96-517, 94 Stat. 3015 (1980) 14

Legislative Materials

Copyright Law Revision: Hearings on H.R. 4347, H.R. 5680, H.R. 6831 and H.R. 6835 before Subcomm. No. 3 of the House Comm. on the Judiciary, 89th Cong., 1st Sess. (1965) 12
Copyright Law Revision: Hearings Pursuant to S. Res. 37 on S. 597 before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on the Judiciary, 90th Cong., 1st Sess. (1967) 12
H.R. Rep. No. 1476, 94th Cong., 2d Sess. (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659 ..... 10, 13, 17
House Comm. on the Judiciary, *Copyright Law Revision* (Comm. Print 1965) (1964 Revision Bill with Discussions and Comments) 12

Other Authorities

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978) 12
Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559 (1994) 15
Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977 (1993) 16
Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* (1994) 5, 15
U.S. Dept. of Commerce, *U.S. Global Trade Outlook 1995-2000: Toward the 21st Century* (1995) 2, 3

TABLE OF AUTHORITIES—Continued

	Page
U.S. Dept. of Commerce, *U.S. Industrial Outlook '94* (1994)	2
Webster's Ninth New Collegiate Dictionary (1987)	15

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF AMICUS CURIAE OF INTELLECTUAL PROPERTY OWNERS IN SUPPORT OF PETITIONER

INTEREST OF AMICUS CURIAE

Intellectual Property Owners ("IPO") files this *amicus curiae* brief in support of the petition of Lotus Development Corporation ("Lotus") for a writ of certiorari to review the ruling of the United States Court of Appeals for the First Circuit.

IPO is a national organization founded in 1972 and representing the interests of owners of intellectual property in the United States. Membership in IPO currently includes nearly one hundred large and mid-size companies and more than three hundred small businesses, universities, independent inventors, authors, executives and attorneys who own or are interested in patents, trademarks, copyrights and other intellectual property rights. Members of

the Board of Directors of the association are listed in the Appendix to this brief. IPO is a non-profit association exempt from federal income tax under Internal Revenue Code § 501(c)(6).

IPO has no interest in either of the parties to this appeal or in the outcome of this appeal, other than to the extent that it affects IPO members and the public generally.[1] Its interest is in the issues before the Court, which are of concern to IPO members, particularly those in the computer and software industries. Many of IPO's members develop computer software, either to compete in the worldwide software market, to complement the services they provide their customers, or to support other segments of their businesses.

This case presents critical issues concerning the extent to which copyright protects computer programs. The computer software industry has long relied on copyright law and the incentives it provides. Under the regime established by Congress that recognizes computer programs as literary works and accords them full protection under copyright law, software development has grown into one of the major American industries. At the same time, the United States has become the dominant force in the world software market, supplying approximately 75% of the nearly $70 billion worldwide packaged software market in 1993 and maintaining a positive balance of trade. U.S. Dept. of Commerce, *U.S. Global Trade Outlook 1995-2000: Toward the 21st Century* 134-35 (1995) ("Trade Outlook"). From 1991 to 1994 the market for packaged software in the United States increased by nearly 41%, to $35.6 billion, *id.*, making it one of the fastest growing sectors in our national economy. U.S. Dept. of Commerce, *U.S. Industrial Outlook '94*, at 27-1 (1994). Employment in the software industry has risen every year

[1] One of IPO's members, International Business Machines Corporation, has recently agreed to purchase all of the outstanding shares of Lotus, a non-member.

since 1988 (the first year for which such data are available), creating nearly 73,000 jobs over that period. *Trade Outlook*, at 134 .

ARGUMENT

The central question in this case is whether copyright in a computer program protects the expression in the program's user interface—*i.e.*, the element of the computer program that permits the program author to communicate with the user, and permits the user to communicate with the computer. At least three courts of appeals have held that expression in the user interface of a computer program is copyrightable.[2] In the decision below,[3] the First Circuit accepted the district court's finding that the user interface before it, the menu command hierarchy [4] of the Lotus 1-2-3 spreadsheet program, contained expression, and that other ways of expressing the ideas underlying the user interface were available. 49 F.3d at 816. The court concluded that this case involves the defendant's "deliberate, literal copying of the Lotus menu command hierarchy." *Id.* at 814. Nonetheless, the court below found that the Lotus 1-2-3 menu command hierarchy was, in its entirety, a "method of operation." For that reason, it held that Section 102(b) of the Copyright Act, 17 U.S.C. § 102(b), deprived it of protection. 49 F.3d at 815.[5]

[2] See *infra*, pp. 4-6.

[3] *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir.), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995) (No. 94-2003).

[4] The Lotus "menu command hierarchy" is the selection of 469 commands and their hierarchical arrangement into more than 50 menus and submenus. 49 F.3d at 809. Together they comprise the heart of Lotus 1-2-3's user interface.

[5] 17 U.S.C. § 102(b) provides:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system,

In the First Circuit's view, " 'method of operation,' as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer." *Id.* Since a user interface is a "means by which a person operates . . . a computer," the court's reasoning appears to preclude protection for any user interface.

But the First Circuit's reasoning has troubling implications for all elements of computer programs—not just user interfaces—and, indeed, for computer programs as a whole. A computer program, defined in the Copyright Act as "a set of statements or instructions to be *used* directly or indirectly in a computer *in order to bring about a certain result*," is a means by which a person operates a computer. 17 U.S.C. § 101 (emphasis supplied). Virtually all elements of a program are means of operating some aspect of a computer. The First Circuit's construction of Section 102(b) might well preclude protection for all such elements, even for the literal code of programs.[6] In his concurring opinion Judge Boudin, in fact, acknowledges that such a literal reading of Section 102(b) "might easily seem to exclude most computer programs from protection." 49 F.3d at 820 (Boudin, J., concurring). That result is clearly at odds with the statute, with the stated intent of Congress and with the weight of judicial authority in other circuits.

---

method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

[6] The court's effort to distinguish "computer code" from the Lotus menu command hierarchy, 49 F.3d at 816, rests on flawed reasoning. See *infra*, note 16.

I. THE DECISION BELOW CONFLICTS WITH DECISIONS OF OTHER CIRCUITS AND CREATES UNCERTAINTY AS TO COPYRIGHT PROTECTION OF COMPUTER PROGRAMS

A. The First Circuit's Holding on Copyrightability of User Interfaces Conflicts With Decisions of Three Other Circuits

The First Circuit's holding is flatly inconsistent with decisions on user interfaces of computer programs in other circuits. Each of the three other courts of appeals that have considered the issue has found that the user interface of a computer program is protectible by copyright. Not one court of appeals has interpreted Section 102(b) of the Copyright Act to preclude protection for the original expression contained in a user interface.[7]

The First Circuit acknowledged that its decision conflicts directly with that of the Tenth Circuit in *Autoskill, Inc. v. National Educational Support Systems, Inc.*, 994 F.2d 1476 (10th Cir.), *cert. denied*, —— U.S. ——, 114 S. Ct. 307 (1993). 49 F.3d at 818-19. In *Autoskill*, the Tenth Circuit found that aspects of the plaintiff's user interface could be protected under copyright. 994 F.2d at 1495 & n.23. The court expressly rejected the defend-

---

[7] We refer, of course, to expression that has not "merged" with the underlying idea. Expression is deemed to have merged with the underlying idea when there is only one way available to express the idea. See generally 3 Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* § 13.03[B][3] (1994) (hereinafter, "Nimmer"). The court below acknowledged, 49 F.3d at 810-11, 816, that there were many alternative ways of expressing the ideas in the Lotus menu command hierarchy (thus ruling out merger), but found that to be "immaterial" to its analysis. *Id.* at 816.

By referring to "original expression" we exclude *"scènes à faire"* —stock treatments of a given subject—that, if protected at all, are protected only against wholesale appropriation. See *Atari Games Corp. v. Oman*, 888 F.2d 878, 886 (D.C. Cir. 1989) (R.B. Ginsburg, J.).

ant's contention, based on Section 102(b), that the means by which a user communicated with the plaintiff's program was a " 'method' not protected by copyright." *Id.* at 1495 n.23.

In *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, —— U.S. ——, 115 S. Ct. 1176 (1995), the Ninth Circuit held that the user interface of the plaintiff's computer program was entitled to copyright protection, though, in that case, only against virtually identical copying. *Id.* at 1439. Since the First Circuit acknowledged that the defendant below, Borland International, Inc. ("Borland") made a virtually identical copy of the menu command structure of Lotus 1-2-3, 49 F.3d at 810, but denied it any protection, the conflict between the First and Ninth Circuits is apparent.[8]

The Fifth Circuit, in *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335 (5th Cir. 1994), *opinion supplemented on denial of rehearing*, 46 F.3d 408 (5th Cir. 1995), held that the entirety of the input and output formats that comprised the user interface of the plaintiff's program was protectible under copyright. 26 F.3d at 1343-44. That court followed Judge Keeton's holding in *Lotus Development Corp. v. Paperback Software, International*, 740 F. Supp. 37 (D. Mass. 1990), upon which Judge Keeton based much of his reasoning in the district court decisions in this case. *E.g., Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78, 89-90 (D. Mass. 1992), *rev'd*, 49 F.3d 807 (1st Cir.), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995) (No. 94-2003).

[8] The First Circuit also recognized that its holding conflicted with what it described as Ninth Circuit dicta in *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465 (9th Cir.), *cert. denied*, —— U.S. ——, 113 S. Ct. 198 (1992), 49 F.3d at 819 n.14.

B. Applied to Computer Programs, the First Circuit's Interpretation of 17 U.S.C. Section 102(b) Conflicts With Decisions of at Least Five Other Circuits

By erecting Section 102(b) as a barrier to deny protection to original expression in computer programs, the First Circuit has set itself in conflict with all circuits that have considered the application of that provision to computer software.

In *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984), the Third Circuit rejected a defendant's claim that Section 102(b) deprives operating system programs (the programs that control the basic functions of a computer, such as the storage and retrieval of information from a magnetic disk) from copyright protection because they implement methods, processes, ideas and systems. *Id.* at 1251-52.[9]

In *M. Kramer Manufacturing Co. v. Andrews*, 783 F.2d 421 (4th Cir. 1986), the Fourth Circuit reversed the district court's holding that plaintiff's computer video game (a type of computer program) was not copyrightable under Section 102(b). The district court's premise had been that plaintiff's work was "a 'system or manner of playing a game' or 'description of a game' and was simply an 'idea . . . .'" *Id.* at 434. The Fourth Circuit held the reliance on Section 102(b) to be "unwarranted" because it failed to recognize "the limited purpose of the subsection itself as stated in its legislative history," *id.*,

[9] The Third Circuit also concluded that the desire to achieve compatibility with a competitor's product does not permit distorting the merger doctrine in order to justify a taking of copyrighted expression. *Apple v. Franklin*, 714 F.2d at 1253. The First Circuit, by contrast, posited Borland's desire to achieve keystroke compatibility with Lotus 1-2-3 as a further ground for its decision that Lotus' menu command hierarchy was unprotectible, justifying Borland's wholesale taking. 49 F.3d at 817-18.

simply to restate the dichotomy between idea and expression. *Id.* at 434-35.

In holding that "methods of operation are not limited to mere abstractions," the decision below conflicts with the Second Circuit's decision in *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992). In *Altai*, the Second Circuit implemented what it described as Section 102(b)'s "fundamental principle . . . that a copyright does not protect an idea, but only the expression of the idea," *id.* at 703, by employing an "abstractions" test as the first step of its infringement analysis.[10] *Id.* at 706. *Altai* applied that test, enunciated by Learned Hand in *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931), to separate protectible expression from unprotectible ideas and other abstractions, for "the purpose of defining the scope of plaintiff's copyright." *Altai*, 982 F.2d at 707 (internal quotes and citation omitted). In contrast to the First Circuit, the Second Circuit did not employ Section 102(b) as a barrier to deny protection to expression embodying ideas and other unprotectible matter. By abstracting, it was able to consider the "core of protectible material" that was separable from matter unprotectible under that section.

The First Circuit's conflict with the Second Circuit's *Altai* decision places it in conflict with those other circuits that have adopted some formulation of an abstractions test. See, e.g., *Engineering Dynamics*, 26 F.3d at 1342-43; *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 834 (10th Cir. 1993). See also, *Atari Games*

[10] The court below stated that its "initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression," 49 F.3d at 816, but recognized that "the *Altai* test would contemplate this being the initial inquiry." *Id.* at 816 n.12. It further acknowledged its conflict with the Second Circuit by criticizing *Altai*'s test because it "may actually be misleading . . . in instructing courts to abstract the various levels . . . ." *Id.* at 815.

Corp. v. Nintendo of America, Inc., 975 F.2d 832, 839 (Fed. Cir. 1992) (approving the *Altai* test in dicta).

C. A Conflict in Rules of Copyrightability Leaves Program Authors Uncertain of Their Rights and Discourages Creation of Computer Programs

In the wake of the First Circuit's decision, there is a profound variation among the circuits in the legal standards governing copyright protection of computer programs. The resulting uncertainty will have a significant adverse impact on the software industry.

The software industry depends on copyright law. Innovation drives the industry, and copyright law provides both the necessary incentive for that innovation and the primary means of protecting the works that result from it. Copyright lies at the very heart of the creative endeavor of software development. And the question of what is copyrightable and what is not lies at the very heart of copyright law.

Moreover, the market for software is not regional, but national (and international). Packaged software such as Lotus 1-2-3 and Quattro Pro (the programs at issue here) is advertised and promoted nationally, and is marketed—directly, through mail-order houses, and through chains of retail stores—in every state in the United States.

As a consequence of these two factors, the First Circuit's decision subjects the industry's primary assets to the uncertainty of flatly contradictory legal rules in the marketplace. A software developer cannot know whether, or to what extent, its investments in creative expression will be protected. Such protection is subject to the vagaries of forum and venue. All that is certain is that the question will arise in the lower courts in every case of alleged copyright infringement involving computer programs.

Software development is expensive. Billions of research and development ("R&D") dollars ride on the disposition

of these issues. Developers will not risk such substantial sums for the time and talent of creative programmers without assurance of sufficient legal protection for the fruits of that investment. Yet developers are now threatened with precisely that risk. There is no way of predicting whether a program or any element of a program will be protected. This uncertainty makes investment of R&D resources in innovative software development an unreasonably risky wager, and will likely force some to invest those resources in other endeavors.

The reverse side of this uncertainty is that software developers cannot predict where the boundaries of permissible copying lie. A program that infringes in Texas may not infringe in Massachusetts. Uncertainty on such fundamental matters defeats the goal of a uniform national copyright law. See H.R. Rep. No. 1476, 94th Cong., 2d Sess., at 129-30 (1976) (the "House Report"), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5745-46 (discussing importance of the goal of a uniform national copyright law, as espoused by the Framers of the Constitution).

Moreover, the decision creates uncertainty in an area of law where decisions in the courts of appeals had been converging. Departing from this trend toward coherence and consensus, the First Circuit has unsettled what had appeared to be settled, and created unnecessary uncertainty about the legal rules that apply to a key national industry.

II. THE DECISION BELOW DENIES COMPUTER PROGRAMS THE COPYRIGHT PROTECTION MANDATED BY CONGRESS AND DIMINISHES THE STATUTORY INCENTIVES TO CREATE THEM

In holding that the district court erred by limiting "Lotus 1-2-3's 'method of operation' to an abstraction," 49 F.3d at 816, the First Circuit erected Section 102(b) of the Copyright Act as an initial barrier, denying protection to certain categories of works, or elements of works,

regardless of their expressive content. The Copyright Act does not read that way, Congress did not intend it to be read that way, and no other court of appeals has read it that way.

The First Circuit's misreading is a direct result of its failure to follow the holdings of this Court and employ appropriate methods of statutory interpretation. As applied to computer programs the First Circuit's holding is not merely wrong: it defeats Congress' clear mandate that computer programs be protected under copyright. The decision departs so substantially from settled copyright law, from accepted means of statutory construction, and from the clearly-stated will of Congress, as to call for an exercise of this Court's power of supervision over the lower Federal Courts.

A. The First Circuit Misinterpreted a Key Provision of the Copyright Act Because It Failed to Employ Appropriate Means of Statutory Construction

Without citing any authority for its conclusion, the court below determined that " 'method of operation,' as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer." 49 F.3d at 815. According to the court, anything that falls within that definition is wholly deprived of copyright protection, even to the extent that it contains creative expression. *Id.* at 815-19.

But a court should interpret a statutory provision in light of "the provisions of the whole law, and . . . its object and policy," *Kelly v. Robinson*, 479 U.S. 36, 43 (1986) (citations and internal quotations omitted), as revealed by the statutory text and legislative history. The First Circuit failed utterly to do this. Without so much as a sideways glance, the First Circuit looked at Section 102(b) in isolation and proffered an interpretation of

the provision that undoes what Congress did elsewhere in the statute.[11]

Congress' inclusion of a definition of "computer program" in Section 101 is, in itself, persuasive evidence of Congress' mandate that computer programs be counted among the "works of authorship" protected under Section 102(a). The legislative history of the Copyright Act confirms that they are.

The Copyright Act of 1976 culminated more than three decades of consideration and debate by Congress. Congress considered many different viewpoints presented in testimony, written submissions, and studies. Protection of computer programs figured prominently in these proceedings,[12] particularly after the Copyright Office—the agency charged with determining that a work sought to be registered is copyrightable subject matter—began issuing registrations for claims of copyright in computer programs in 1964. *Final Report of the National Commission on New Technological Uses of Copyrighted Works*, 15 (1978) ("CONTU Final Report").

---

[11] The canon that a court "construe[s] a statutory term in accordance with its ordinary or natural meaning," *FDIC v. Meyer*, 510 U.S. ——, 114 S. Ct. 996, 1001 (1994), should not be applied to a single phrase in isolation in a way that defeats other provisions of the statute.

[12] *See, e.g., Copyright Law Revision: Hearings on H.R. 4347, H.R. 5680, H.R. 6831 and H.R. 6835 before Subcomm. No. 3 of the House Comm. on the Judiciary*, 89th Cong., 1st Sess. 1144-50 (Statement of John F. Banzhaf III), 1898-99 (letter from Graham W. McGowan to Hon. Edwin E. Willis, House Judiciary Committee), 1428, 1443, 1446-48, (statement of Bella L. Linden), 1472 (statement of Howard A. Meyerhoff), 1565 (statement of Fred S. Siebert) (1965); 5 House Comm. on the Judiciary, *Copyright Law Revision* 62-63 (testimony of Morton David Goldberg), 271 (statement of H. R. Mayers) (Comm. Print 1965) (1964 Revision Bill with Discussions and Comments); *Copyright Law Revision: Hearings Pursuant to S. Res. 37 on S. 597 before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on the Judiciary*, 90th Cong., 1st Sess. 192-201 (1967) (testimony and statement of Prof. Arthur R. Miller).

Still, Congress sought further input and constituted a panel of experts, the National Commission on New Technological Uses of Copyrighted Works ("CONTU"), to study and make recommendations concerning, *inter alia*, the proper regime for protecting computer programs. *Pub.* L. No. 93-573 (1974). The Copyright Act was enacted before CONTU's study was complete. However, the committee report that Congress prepared when deliberating on the Copyright Act [13] states plainly that computer programs are among the literary works protected by Section 102(a)(1) of the statute:

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.

House Report, at 54, *reprinted in* 1976 U.S.C.C.A.N. at 5667.

CONTU took testimony and written submissions from numerous experts in copyright and information technology. *CONTU Final Report*, at 113-14, 121-29. After consideration and debate, CONTU concluded that copyright should protect computer programs to the same degree as, and under the same principles that apply to, other copyrighted works. *CONTU Final Report*, at 11-14. In 1980 Congress adopted CONTU's recommendations with immaterial changes—including CONTU's proposed

---

[13] See *Gustafson v. Alloyd Co.*, —— U.S. ——, 115 S. Ct. 1061, 1072 (1995) ("If legislative history is to be consulted, it is preferable to consult the documents prepared by Congress when deliberating.").

definition of "computer program."[14] Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980).

Under the First Circuit's construction of Section 102(b), however, it is questionable whether *any* protection remains for computer programs.[15] Notwithstanding the court of appeals' unpersuasive attempt to limit the reach of its holding,[16] its interpretation of Section 102(b) conflicts directly with Congress' intent that computer programs be protected by copyright.

---

[14] As this brief chronology demonstrates, copyright protection for computer programs is neither an historical accident, nor an impulsive act that received little attention by the legislators. Copyright protection for computer programs is the result of decades of open debate and careful consideration.

[15] See *supra*, p. 3.

[16] The First Circuit's attempt to limit its holding by distinguishing "computer code" from the Lotus menu command hierarchy is based on flawed and unpersuasive reasoning. The court stated that the menu command hierarchy of Lotus 1-2-3 was a method of operation because "[w]ithout the menu command hierarchy, users would not be able to access and control, or indeed make use of, *Lotus 1-2-3's* functional capabilities." 49 F.3d at 815 (emphasis supplied). On the other hand, the code of Lotus 1-2-3 is not a method of operation because, in order for a competing program "to offer *the same capabilities as Lotus 1-2-3* [a competitor does not] have to copy Lotus' underlying code . . . ." *Id.* at 816 (emphasis supplied). The court asked a different question regarding the menu command hierarchy (whether it was needed to operate *a particular program*—Lotus 1-2-3) than it asked regarding code (whether the particular code was needed to offer *the same capabilities* as Lotus 1-2-3). Had it applied to the user interface the same standard it applied to the code, the Court of Appeals would have found the menu command hierarchy copyrightable, since a program offering *the same capabilities* as Lotus 1-2-3 could have a different menu command hierarchy, as the district court found. By the same token, had it applied to the program code the same standard it applied to the user interface, it would have recognized the flaw in its reasoning. The code of Lotus 1-2-3 is necessary to operate Lotus 1-2-3, as the code of any program is necessary to operate that particular program, making the code—under the First Circuit's reasoning—an unprotectible method of operation.

The phrase "method of operation" is not defined in the Copyright Act. The court below, presumably adopting what it viewed as the common use of those words, 49 F.3d at 821 (Boudin, J., concurring), constructed a definition similar to that adopted by Congress for a "computer program" in Section 101. On its face, that construction must be wrong. It cannot be presumed that Congress intended to adopt for uncopyrightable matter a definition that has essentially the same meaning as the statutory definition of a protectible work of authorship—namely, a computer program.[17]

But even assuming that the First Circuit's definition were defensible, that would not be dispositive of the question whether the manner in which a "method of operation" is expressed is copyrightable. Nothing in Section 102(b) states or even suggests that *expression* that is separable from (*i.e.*, is not merged with) a "method of operation" is uncopyrightable. Separating a method of operation from expression is analytically no different from separating ideas (another category listed in Section 102(b)) from expression.[18]

The First Circuit not only failed to consider the whole statute, but even failed to consider the whole subsection.

---

[17] See 1 *Nimmer*, at § 2.03[E]; Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559, 2570 (1994) ("[S]ince we know that Congress did determine in 1980 to protect computer programs, the terms 'process,' 'system,' or 'method of operation' must not be understood literally." (footnote omitted)).

[18] The fallacy of the First Circuit's construction is illustrated by substituting "idea" for "method of operation." "Idea" can be defined as "a formulated thought or opinion." *Webster's Ninth New Collegiate Dictionary* 597 (1987). Therefore (following the First Circuit's logic) any element of a work that fits that definition is unprotectible, no matter how expressed. That interpretation would, of course, disqualify most writings from copyright protection.

Section 102(b) denies protection to "any idea, procedure, process, system, method of operation, concept, principle, or discovery . . . ." 17 U.S.C. § 102(b). The presence of "idea" at the head of this list of unprotectible matter itself strongly suggests that Congress intended Section 102(b) to restate the "idea expression" dichotomy, first enunciated by the Court in *Baker v. Selden*, 101 U.S. 99 (1879).[19] Absent merger (another principle that owes its origin to *Baker*), the idea expression dichotomy does not deny protection to expression in a work.[20] The same is true of each of the other items listed in Section 102(b).

Again, the legislative history of the Copyright Act confirms this reading:

> Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law.
>
> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged.

[19] The final phrase of Section 102(b) ("regardless of the form in which it is described, explained, illustrated, or embodied in such work") also supports the interpretation of that provision as a statement of the idea/expression dichotomy. The statutory language clearly contemplates that ideas, procedures, processes and the like are separable from the form in which they are "described, explained, illustrated, or embodied . . . ." Thus a dichotomy exists between the unprotectible matter and its "form" or expression.

[20] *See* Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 986 (1993) ("Since *Baker*, courts have reaffirmed repeatedly that functionality poses no per se bar to copyrightability.") (the author was a member of CONTU who served on its software subcommittee).

House Report, at 57, 1976 U.S.C.C.A.N. at 5670. See also *id.* at 54, 1976 U.S.C.C.A.N. at 5667 (computer programs are protectible under Section 102(a)(1) "to the extent that they incorporate authorship in the programmer's expression . . . as distinguished from the ideas themselves.").

Implementing Congress' express statement in the legislative history of the Copyright Act, courts of appeals have universally found Section 102(b) to be a restatement of the idea expression dichotomy. See, e.g., *CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc.*, 44 F.3d 61, 69 & n.12 (2d Cir. 1994), *petition for cert. filed*, 63 U.S.L.W. 3892 (U.S. June 7, 1995); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); *M. Kramer Mfg. Co., Inc. v. Andrews*, 783 F.2d 421, 434 (4th Cir. 1986); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 533 & n.8 (5th Cir.), *cert. denied*, —— U.S. ——, 115 S. Ct. 82 (1994); *Robert R. Jones Assocs., Inc. v. Nino Homes*, 858 F.2d 274, 277 (6th Cir. 1988); *Toro Co. v. R & R Prods. Co.*, 787 F.2d 1208, 1211-12 (8th Cir. 1986); *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 & n.11 (9th Cir. 1994); *Autoskill Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 (10th Cir. 1993); *Cable/Home Communication Corp. v. Network Prods., Inc.*, 902 F.2d 829, 842-43 (11th Cir. 1990). See also, *Atari, Inc. v. North American Philips Consumer Elecs. Corp.*, 672 F.2d 607, 615 (7th Cir.) ("The Copyright Act of 1976 codifies this idea-expression dichotomy. 17 U.S.C. § 102(b)"), *cert. denied*, 459 U.S. 880 (1982).

B. Denying Protection to Expression in Computer Programs Erodes Severely the Incentives Congress Enacted for Their Authors

The Constitution grants Congress the power "To Promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive

Right to their respective Writings and Discoveries," U.S. Const., art. I, § 8, cl. 8. It thus establishes both the underlying goal of copyright law (to promote the progress of science (knowledge)) and the mechanism for achieving that goal (granting authors exclusive rights for a limited period of time).

The Court has confirmed that the public good is served by providing authors with these exclusive rights as an incentive to create works of authorship:

> The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in "Science and useful Arts." Sacrificial days devoted to such creative activities deserve rewards, commensurate with the services rendered.

Mazer v. Stein, 347 U.S. 201, 219 (1954).

The computer software industry relies on these incentives, and they have caused the industry to flourish and grow. Any fundamental alteration in the balance of these incentives, as in the decision below, will have a substantial and detrimental effect on the industry.

Software development (including user interface design)[21] requires a substantial investment in creative expression—the very creativity that the economic incentives in the Copyright Act are designed to protect and nurture. Eliminating copyright protection for programs or expressive elements of programs undermines those incentives to innovate.[22] Even casting substantial doubt on the availability and extent of such protection stifles progress and creativity, and disserves the goals of the Copyright Act.[23]

[21] As the most visible element of a computer program, the user interface is also the most easily plagiarized. Excluding this important programming element from protection will inevitably lead to more unauthorized copying.

[22] In *Apple Computer, Inc. v. Formula Int'l, Inc.*, 562 F. Supp. 775, 783 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984), the court observed: "Few companies are going to invest the time and resources to develop new programs if their products can be freely duplicated by anyone. Such 'competitors,' who could undersell the originator simply because they don't have its development costs, would destroy the market which any innovator needs to recoup his investment."

C. The First Circuit's Construction of Section 102(b) Is Such a Significant Departure From That of the Overwhelming Majority of Courts of Appeals and District Courts as to Call for an Exercise of This Court's Power of Supervision

The First Circuit's approach to Section 102(b) of the Copyright Act is a substantive and jurisprudential anomaly. The Court of Appeals, considering itself to be "navigating in uncharted waters," 49 F.3d at 813, weighed anchor and set itself adrift doctrinally. The court failed to follow the course that Congress charted for the judiciary in the Copyright Act and its legislative history. It ignored the overwhelming number of other courts that have recognized Section 102(b) as a restatement of the idea/expression dichotomy. And in so doing it reached

[23] This is so despite the First Circuit's mistaken belief that its holding would foster progress in the sofeware development industry. 49 F.3d at 817-18. The First Circuit was obliged to abide by the balance struck by Congress in establishing the scheme of incentives for authors. This Court's admonition to another court of appeals is appropriate in this instance as well:

> We agree with the Court of Appeals that copyright is intended to increase and not impede the harvest of knowledge. But we believe the Second Circuit gave insufficient deference to the scheme established by the Copyright Act for fostering the original works that provide the seed and substance of this harvest. The rights conferred by copyright are designed to assure contributors to the store of knowledge a fair return for their labors.

Harper & Row, Publishers v. Nation Enters., 471 U.S. 539, 545-46 (1985) (citation omitted).

erroneous conclusions that subvert the goals of the Copyright Act and jeopardize the ability of a critical industry to continue its significant contributions to our nation's economy.

The issues raised in this case are as ubiquitous as they are important. The question of copyrightability arises in all copyright infringement cases, and is central to many. The substantial degree to which the First Circuit has lost its way in this case calls for an exercise of the Court's power of supervision over the federal judiciary. Cf. *Florida v. Rodriguez*, 469 U.S. 1, 7 (1984) (Stevens, J., dissenting) ("As the Court of last resort in the federal system, we have supervisory authority and therefore must occasionally perform a pure error-correcting function in federal litigation.").

CONCLUSION

The petition for a writ of certiorari should be granted.

Respectfully submitted,

MORTON DAVID GOLDBERG
GARY L. GRISWOLD
HERBERT C. WAMSLEY *
INTELLECTUAL PROPERTY OWNERS
1255 Twenty-Third Street, N.W.
Suite 850
Washington, D.C. 20037
(202) 466-2396
Attorneys for Amicus Curiae
* Counsel of Record

July 7, 1995

APPENDIX

MEMBERS OF BOARD OF DIRECTORS OF INTELLECTUAL PROPERTY OWNERS *

Robert L. Andersen
FMC Corporation

Norman L. Balmer
Union Carbide Corporation

Erwin F. Berrier, Jr.
General Electric Company

W. Dexter Brooks
Coca-Cola Company

James L. Fergason
Optical Shields, Inc.

Michael W. Glynn
Ciba

Stephen S. Grace
Dow Chemical Company

Gary L. Griswold
3M

Robert P. Hayter
United Technologies Corp.

Edward P. Heller, III
Seagate Technology, Inc.

Wayne C. Jaeschke
Henkel Corporation

Geoffrey M. Karny
Genetic Therapy, Inc.

John J. Klocko, III
E.I. DuPont de Nemours & Co.

William E. Lambert, III
Rohm & Haas Company

Edward T. Lentz
SmithKline Beecham Corporation

William F. Marsh
Air Products & Chemicals, Inc.

Roger L. May
Ford Motor Company

John P. McDonnell
AT&T

Alexander J. McKillop
Mobil Oil Corporation

Robert E. Muir
Caterpillar Inc.

Lita Nelsen
Massachusetts Institute of Technology

Wallace L. Oliver
Amoco Corporation

Garo A. Partoyan
Mars Incorporated

Harold R. Patton
Medtronic, Inc.

Jacobus C. Rasser
Procter & Gamble

---

* IPO procedures require approval of positions in briefs by a three-fourths majority of directors present and voting.

Peter C. Richardson
Pfizer, Inc.

Allen W. Richmond
Phillips Petroleum Company

Melvin J. Scolnick
Pitney Bowes Inc.

Victor Siber
IBM Corporation

Roger S. Smith
Morgan & Finnegan

Howard C. Stanley
Monsanto

Gary D. Street
Hoechst Marion Roussel

Herbert C. Wamsley
Intellectual Property Owners

Ogden H. Webster
Eastman Kodak Company

John K. Williamson
Westinghouse Electric Corp.

FILED
JUN 7 1995

No. 94-2003

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL,

Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

MOTION OF INFORMATION TECHNOLOGY INDUSTRY COUNCIL FOR LEAVE TO FILE BRIEF *AMICUS CURIAE* IN SUPPORT OF THE PETITION FOR A WRIT OF CERTIORARI AND BRIEF *AMICUS CURIAE*

JOHN F. COONEY
Counsel of Record
KENNETH C. BASS, III
JOHN G. MILLIKEN
VENABLE, BAETJER, HOWARD & CIVILETTI, LLP
1201 New York Avenue, N.W.
Suite 1000
Washington, D.C. 20005-3917
(202) 962-4800

Counsel for Information Technology Industry Council

July 7, 1995

PRESS OF BYRON S. ADAMS, WASHINGTON, D.C. 1-800-347-8208

BEST AVAILABLE COPY

No. 94-2003

IN THE
SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

MOTION OF INFORMATION TECHNOLOGY INDUSTRY COUNCIL FOR LEAVE TO FILE BRIEF AMICUS CURIAE IN SUPPORT OF THE PETITION FOR A WRIT OF CERTIORARI AND BRIEF AMICUS CURIAE

The Industry Technology Industry Council ("ITI") respectfully moves for leave to file the attached Brief *amicus curiae* in support of the Petition for a Writ of Certiorari. The consent of the attorney for the Respondent was requested but refused.

The interest of ITI in this case arises from its representation of the leading United States providers of information technology products and services. Most of its members author and publish computer programs to license in the open market. ITI's membership comprises a major segment of the United States economy. In 1992, their revenues were over $277 billion worldwide. They employ over one million people in the United States, a significant number of whom write computer programs. They also fund over 16% of all research and development in the United States, much of which involves authoring creative computer programs.

It is believed that the Brief which amicus curiae is requesting permission to file will contain a more complete description of the importance of the decision of the First Circuit for the software industry, a field which has been marked by rapid innovation, and in which the United States currently leads the world. Much of that innovation involves advances in the user interfaces, such as the decision tree hierarchy deemed unprotectable by the court of appeals. However, innovation and enhancement of the interface is critical to promoting the user friendliness of programs, and thus to the realization of the enormous efficiencies offered by the widespread distribution of personal computers to large numbers of non-technically trained people.

ITI believes that the Court's decision that user interfaces are unprotectable under 17 U.S.C. § 102(b), and that one may freely copy elements of a program interface designed by others, has significant implications for the degree of copyright protection to be afforded United States software in this country and other countries, and thus affects the international competitiveness of this industry. The Petition raises an important legal question that warrants review by the Court at this time.

Respectfully submitted,

_____

Kenneth C. Bass, III

Venable, Baetjer, Howard & Civiletti, LLP
1201 New York Ave.
Suite 1000
Washington, D.C. 20005-3917
(202) 962-4800

Counsel for
Information Technology
Industry Council

July 7, 1995

Table of Contents

Page No.

Interest of *Amicus Curiae* 1

Reasons for Granting the Petition 3

I. This Case Presents a Legal Issue of Great Practical Significance to the Computer Software Industry 4

II. The Decision of the First Circuit Conflicts with The Decisions of Other Courts of Appeals 9

III. The Decision Below Creats Uncertainty About the Application of the Copyright Laws to Computer Programs 10

Conclusion 12

Table of Authorities

Page No.

CASES

Autoskill, Inc. v. National Educational Support Systems, Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 S. Ct. 307 (1993) 9

Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir. 1992) 7

Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 9

Digital Communications Associates, Inc. v. Softklone Distributing Corp., 659 F. Supp. 449 (N.D. Ga. 1987) 7

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1344 (5th Cir. 1994) 9

Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984 (D. Conn. 1989) 7

Telemarketing Resources v. Symantec Corp., 12 U.S. P.Q.2d 1991 (1990) 7

STATUTES

17 U.S.C. § 102(b) 2, 4, 10

19 U.S.C. § 2412(B)(2)(a) 6

Pub. L. No. 103-465, § 101(d)(15), 108 Stat. 4809, 4815 6

Pub. L. No. 94-553, 90 Stat. 2541 (1976) 6

Pub. L. No. 96-517, 94 Stat. 3015 (1980) 6

OTHER AUTHORITIES

D. Nimmer & M. Nimmer, 3 *Nimmer on Copyright*, § 13.03[F][1] 9

No. 94-2003

IN THE
SUPREME COURT OF THE UNITED STATES
OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Petition for a Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF OF AMICUS CURIAE INFORMATION TECHNOLOGY INDUSTRY COUNCIL IN SUPPORT OF THE PETITION FOR A WRIT OF CERTIORARI

INTEREST OF *AMICUS CURIAE*

The Industry Technology Industry Council ("ITI") submits this Brief as *amicus curiae* in support of the Petition for a Writ of Certiorari.

ITI represents the leading United States providers of information technology products and services. Most of its members author and publish computer programs to license in the open market. ITI's membership comprises a major segment of the United States economy. In 1992, their revenues were over $277 billion worldwide. They employ over one million people in the United States, a significant number of whom write computer programs. They also fund over 16% of all research and development in the United States, much of which involves authoring creative computer programs.

ITI submits this Brief *amicus curiae* in order to emphasize the importance of the decision of the First Circuit for the software industry, a field which has been marked by rapid innovation, and in which the United States currently leads the world. Much of that innovation involves advances in the user interfaces, such as the decision tree hierarchy deemed unprotectable by the court of appeals. However, innovation and enhancement of the interface is critical to promoting the user friendliness of programs, and thus to the realization of the enormous efficiencies offered by the widespread distribution of personal computers to large numbers of non-technically trained people. ITI believes that the Court's decision that user interfaces are unprotectable under 17 U.S.C. § 102(b), and that one may freely copy elements of a program interface designed by others, raises a critical legal question that warrants review by the Court at this time.

REASONS FOR GRANTING THE PETITION

The question whether the user interface of a software program may be copyrighted is a critical legal question of great practical significance for one of the most innovative and internationally competitive industries in the United States. This is an issue of law of broad applicability. Litigation about the copyrightability of user interfaces has been before the courts of appeals for several years and will continue to arise until this question is finally resolved by the Court. The issue is well framed for resolution in this case, because there is no factual dispute whether the respondent in fact copied petitioner's user interface. Further, the First Circuit has adopted an unambiguous rule that decision tree hierarchies in a user interface are not copyrightable as a matter of law under any circumstances.

There is a conflict in approach among the courts of appeals as to whether the user interface portion of a software program may be copyrighted. Software programs are sold in a national market. Further, programs are uniquely portable, in that they may be shipped from one end of the country to another in a few seconds over telephone lines, and software development efforts can readily be shifted from one location to another to take advantage of any disparities in legal rules. Authors of software cannot conduct their business efficiently under the current state of confusion, where there are different rules in different circuits as to the copyrightability of one of the most critical features of computer programs. The importance of this issue justifies granting certiorari at this time.

Finally, the rule adopted by the First Circuit will introduce uncertainty in the law and generate further litigation over software copyright issues. The court below held that the decision tree hierarchy of the user interface is not copyrightable under Section 102(b) of the Copyright Act, 17 U.S.C. § 102(b), because it constitutes a "method of operation", a set of instructions to a machine. But virtually every aspect of software, in both the user interface and other portions of the program, is an instruction to the computer. The distinction drawn by the First Circuit contains no limiting principle by which to differentiate which instructions in a program are copyrightable from those which are not. Accordingly, this rule gives little guidance to authors of software about the scope of their rights. And efforts to apply it are likely to result in much further litigation unless the Court resolves the issue at this time.

I. THIS CASE PRESENTS A LEGAL ISSUE OF GREAT PRACTICAL SIGNIFICANCE TO THE COMPUTER SOFTWARE INDUSTRY

The court of appeals has decided an important legal question that is of great significance to the computer software industry and warrants review by this Court. The question whether the user interface of a computer program is, as a matter of law, copyrightable subject matter is a recurrent legal issue that will continue to arise until finally resolved by this Court. Resolution of this issue is critical to the development of the software industry, both in the United States and internationally.

A. The user interface is one of the most important aspects of a computer program.

It is a shorthand expression for those elements of the program that permit its author to communicate with the user and that permit the user to communicate with the computer. A substantial degree of innovation and creativity are involved in determining how to design the user interface, with the goal of making it "user friendly" -- that is, to make it easier for users of different types of intelligence and experience to understand and utilize various features of the program. The design of the user interface has become increasingly important with the distribution of personal computers to large numbers of non-technically trained users. While the engineering efficiency of the actual applications remains an important technical attribute of a computer, there is a significant art involved in making a program accessible to non-engineers, so that large numbers of average citizens may take advantage of the enormous efficiencies offered by computer technology.

B. Software development is one of the most dynamic industries in the United States economy. United States companies are the acknowledged world leader in the development and marketing of software, especially for the off-the-shelf commercial software that has fueled the revolutionary growth in information technology.

Development of successful software in today's highly competitive market requires a substantial degree of creativity and a significant investment of capital. The software industry depends upon the intellectual property protection afforded by the copyright laws, both to create the incentives to encourage authors to develop new products and to prevent

others from copying the resulting software and improperly diverting the rewards. Denying copyright protection to the fruits of this intense research and development effort, including user interfaces that effectively communicate with large numbers of potential users, will have an adverse impact on this vital industry.

The Federal government has long recognized the importance of providing adequate copyright protection to authors of innovative computer programs. In 1976 and 1980, Congress amended the Copyright Act for the explicit purpose of providing greater copyright protection to computer programs, in order to stimulate greater innovation.[1]/ Since the mid-1980s, both Congress and the President have made sustained efforts to make certain that other countries, especially those in the Pacific Basin, respect copyrights on United States software and do not permit their citizens to pirate the work of this important domestic industry.

The most visible aspects of this program have been the investigations and actions taken by the United States under the "Special 301" provision of the Omnibus Trade and Competitiveness Act of 1988 (19 U.S.C. § 2412(b)(2)(A)). These intense bilateral activities have been complemented by the multilateral Uruguay Round Agreement on the Trade Related Aspects of Intellectual Property Rights, ratified by Congress in 1994 (Pub. L. No. 103-465, § 101(d)(15), 108 Stat. 4809, 4815). Under this Agreement, computer software will receive the highest level of copyright protection available.

1/ Pub. L. No. 94-553, 90 Stat. 2541 (1976); Pub. L. No. 96-517, 94 Stat. 3015 (1980).

Any elements of software that are not protected by copyright in this country are highly unlikely to be protected abroad. Stated another way, if aspects of a software program can be copied freely in this country, then they will be duplicated by software developers in other countries. Thus, the decision below, denying copyright protection as a matter of law to the decision tree hierarchy of a user interface of one popular program, will have significant repercussions on the degree of international competition faced by the United States software industry, in addition to its effects in the domestic market.

C. The First Circuit based its holding on the threshold ground that these elements of the program are not copyrightable because they constitute a "method of operation" -- that is, "the means by which users control and operate Lotus 1-2-3" (Pet. App. at 15a). The decision thus creates a legal question of general applicability, which extends well beyond this particular program. The question of whether the user interface of a program may be copied freely is a recurrent issue that is virtually certain to keep arising until the matter ultimately is resolved by this Court.[2]/

The issue is well framed for resolution by the Court in this case.

--First, the most vexing question in copyright infringement cases, whether the defendant actually copied

---

2/ *See, e.g., Digital Communications Associates, Inc. v. Softklone Distributing Corp.*, 659 F. Supp. 449 (N.D. Ga. 1987); *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F. Supp. 984 (D. Conn. 1989); *Telemarketing Resources v. Symantec Corp.*, 12 U.S. P.Q.2d 1991 (1990); *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465 (9th Cir. 1992).

the expression of the plaintiff, is not at issue here. The First Circuit found that the case involves "Borland's deliberate, literal copying of the Lotus menu command hierarchy" (Pet. App. at 14a).

> "[W]e are faced only with whether the Louts menu command hierarchy is copyrightable subject matter in the first instance, for Borland concedes that Louts has a valid copyright in Lotus 1-2-3- as a whole and admits to factually copying the Lotus menu command hierarchy." (Id. at 11a (footnote omitted)).

--Second, the First Circuit has adopted a clearly defined legal position, that under no circumstances may copyright protection extend to the aspects of a user interface that "provide[] the means by which users control and operate" the program. (Id. at 15a). Accordingly, this case presents the underlying legal issue in a context that would facilitate its resolution by this Court.

II. THE DECISION OF THE FIRST CIRCUIT CONFLICTS WITH THE DECISIONS OF OTHER COURTS OF APPEALS.

The First Circuit stated that it was navigating in uncharted waters and that it knew of "no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue)." (Pet. App. 12a). At another point, however, the court acknowledged that its "holding that methods of operation are not limited to abstractions goes against" *Autoskill, Inc. v. National Educational Support Systems, Inc.*, 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 S. Ct. 307 (1993). (Pet. App. 21a). The Petition discusses at some length the differences in the legal standard applied by the First Circuit and by other Circuits, based on the approach articulated in the Nimmer treatise. (Pet. at 23-26).[3/]

We believe that there is a sufficient conflict in approach among the courts of appeals that, given the importance of the issue, justifies granting certiorari at this time. Computer software is sold in a single nationwide market. It is uniquely portable and could be transmitted from place to place electronically, in order to take advantage

---

3/ D. Nimmer & M. Nimmer, 3 *Nimmer on Copyright*, § 13.03[F][1]. *See Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992); *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1344 (5th Cir. 1994)("if a best-selling program's interface were not copyrightable, competitors would be free to emulate the popular interface exactly so long as the underlying programs were not substantially similar. This cannot be the law.").

of differences in legal rules from jurisdiction to jurisdiction. Software development requires uniform national standards on the core legal issue of the standards that determine the degree of copyright protection available. This dynamic industry cannot function effectively if the legal rules governing copyrightability of critical aspects of programs differ from circuit to circuit.

Accordingly, the Court should grant certiorari to resolve the conflict among the circuits concerning the legal standards determining the protectability of user interfaces under the Copyright Act.

III. THE DECISION BELOW CREATES UNCERTAINTY ABOUT THE APPLICATION OF THE COPYRIGHT LAWS TO COMPUTER PROGRAMS.

The basis on which the First Circuit articulated its decision creates uncertainty about what elements, if any, of a computer program are copyrightable as a matter of law.

The court of appeals concluded that the menu command hierarchy of the user interface of this program is a "method of operation", which is not copyrightable pursuant to Section 102(b) of the Copyright Act. The court stated that a "method of operation" within the meaning of that provision "refers to the means by which a person operates something, whether it be a car, a food processor, or a computer". (Pet App. at 15a). It further reasoned that the "Lotus command hierarchy provides the means by which users control and operate Lotus 1-2-3". (Id.).

Virtually every aspect of a computer program is an instruction to a machine. Thus, almost every aspect of a program is potentially susceptible to being characterized as a "method of operation" under the First Circuit's opinion. Further, the court's rule contains no limiting principle by which to differentiate those aspects of a program that are copyrightable from those that are not protectable.

Unless the issue is resolved by this Court, the articulation of the legal standard adopted by the court of appeals will generate litigation concerning whether a particular instruction in a program is to be deemed a "means by which users control and operate" the program, which would not be protectable, as opposed to a means by which the consumer operates the computer, which presumably would still be protectable. This important legal question warrants review by this Court.

CONCLUSION

For the reasons set forth above, the petition for a writ of certiorari should be granted.

Respectfully submitted,
John F. Cooney
(Counsel of Record)
Kenneth C. Bass, III
John G. Milliken

Venable, Baetjer, Howard &
Civiletti, LLP
1201 New York Ave.
Suite 1000
Washington, D.C. 20005-3917
(202) 962-4800

Counsel for Amicus Curiae,
Information Technology
Industry Council

July 7, 1995

BEST AVAILABLE COPY

No. 94-2003

Supreme Court, U.S.
FILED
NOV 9 1995
CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

JOINT APPENDIX
VOLUME I OF II
(Pages J.A. 1 to J.A. 539)

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

GARY L. REBACK
Counsel of Record
MICHAEL BARCLAY
SUSAN A. CREIGHTON
COLLEEN BAL
WILSON, SONSINI, GOODRICH & ROSATI
Professional Corporation
650 Page Mill Road
Palo Alto, California 94304
(415) 493-9300

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY & GELHAAR
One Post Office Square
Boston, Massachusetts 02109-2105
(617) 542-7400

Counsel for Respondent

(Petitioner's counsel continued on inside front cover)

PETITION FOR CERTIORARI FILED JUNE 7, 1995
CERTIORARI GRANTED SEPTEMBER 27, 1995

553 pp

NEAL D. GOLDMAN
LOTUS DEVELOPMENT
CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-1127

DONALD J. ROSENBERG
International Business Machines
Corporation
44 South Broadway
White Plains, New York 10601
(914) 288-4010

Counsel for Petitioner

TABLE OF CONTENTS

PAGE

PLEADINGS

Docket Entries ..... J.A. 1

Complaint, filed July 2, 1990 (Dkt. No. 1) ..... J.A. 13

Answer and Demand for Jury Trial, filed July 31, 1990 (Dkt. No. 4) ..... J.A. 21

Brief Amicus Curiae On Behalf Of Register Of Copyrights, filed September 30, 1991 (Dkt. No. 85) ..... J.A. 27

Supplemental Complaint, filed January 19, 1993 (Dkt. No. 305) ..... J.A. 32

Answer to Supplemental Complaint, filed January 21, 1993 (Dkt. No. 307) ..... J.A. 36

Stipulation and Order Regulating Trial, filed February 3, 1993 (Dkt. No. 330) ..... J.A. 40

Borland's Motion To Amend Original Answer, filed February 26, 1993, with Proposed Amended Answer annexed thereto (Dkt. No. 341) ..... J.A. 56

Notice of filing of Trial Record, filed May 5, 1993 (Dkt. No. 359) ..... J.A. 64

Notice of filing of "Software Library," filed May 5, 1993 (Dkt. No. 360) ..... J.A. 68

PAGE

HEARING & TRIAL TRANSCRIPTS

Lotus v. Paperback Trial Transcript, dated March 22, 1990, Vol. XII (Submitted with Dkt. No. 200) ...J.A. 71

Conference, dated June 18, 1991 (Dkt. No. 66) .....J.A. 74

Memorandum and Order, dated March 20, 1992 (Dkt. No. 164) .....J.A. 77

Hearing on Cross-Motions for Summary Judgment, dated May 19, 1992 (Dkt. No. 194) .....J.A. 80

Procedural Order with Exhibits thereto, dated September 30, 1992 (Dkt. No. 232) .....J.A. 86

Non-Jury Trial Proceedings Day 3, dated February 3, 1993 (Dkt. No. 333) .....J.A. 106

Non-Jury Trial Proceedings Day 4, dated March 31, 1993 (Dkt. No. 358).....J.A. 157

Non-Jury Trial Proceedings Day 5, dated April 1, 1993 (Dkt. No. 404).....J.A. 279

Non-Jury Trial Proceedings Day 6, dated April 2, 1993 (Dkt. No. 405).....J.A. 301

AFFIDAVITS & DECLARATIONS
(Submitted with Dkt. No. 359 unless noted otherwise)

Declaration of Timothy A. Boyd, dated September 26, 1991 (Submitted with Dkt. No. 89) .....J.A. 307

PAGE

Declaration of Eugene K. Buechele, dated September 27, 1991 .....J.A. 310

Second Declaration of Eugene K. Buechele, dated August 27, 1992 (Submitted with Dkt. No. 199) J.A. 313

Declaration of Jennifer Dos Remedios, dated November 1, 1991.....J.A. 317

Declaration of James C. Emery, dated November 1, 1991.....J.A. 320

Declaration of Bernard A. Galler in *Lotus v. Paperback*, dated November 27, 1989.....J.A. 357

Second Declaration of Bernard A. Galler, dated November 1, 1991.....J.A. 382

Declaration of Marvin Goldschmitt, dated September 24, 1991 (Submitted with Dkt. No. 89) .....J.A. 399

Declaration of David E. Kieras, dated September 26, 1991.....J.A. 407

Declaration of Robert H. Kohn, dated September 27, 1991.....J.A. 419

Supplemental Declaration of Robert H. Kohn, dated January 25, 1993 .....J.A. 428

Declaration of Thomas M. Lemberg, dated May 6, 1991.....J.A. 435

Declaration of David E. Liddle, dated September 27, 1991.....J.A. 439

PAGE

Declaration of Jim P. Manzi, dated November 1, 1991 ..... J.A. 470

Declaration of Arthur R. Miller, dated January 3, 1985 (Addendum to Borland's Reply Brief before the First Circuit) ..... J.A. 476

Second Declaration of Arthur R. Miller, dated May 20, 1985 (Addendum to Borland's Reply Brief before the First Circuit) ..... J.A. 487

Declaration of Judith S. Olson, dated September 26, 1991 ..... J.A. 498

Declaration of Vern L. Raburn, dated September 22, 1991, with Exhibits A and B annexed thereto (Submitted with Dkt. No. 89) ..... J.A. 525

Affidavit of Jonathan M. Sachs in *Lotus v. Paperback*, dated November 27, 1989 (Submitted with Dkt. No. 36) ..... J.A. 537

DEPOSITIONS
(Submitted with Dkt. No. 359 unless noted otherwise)

Deposition of Adam Bosworth, April 29, 1991 ..... J.A. 540

Deposition of Adam Bosworth, April 30, 1991 ..... J.A. 567

Deposition of Robert H. Dickerson, September 12, 1991 ..... J.A. 592

Deposition of James Emery, November 8, 1991 ..... J.A. 597

PAGE

Deposition of Bernard A. Galler in *Lotus v. Paperback*, January 25, 1990 (Submitted with Dkt. No. 92) ..... J.A. 602

Deposition of Bernard A. Galler, August 28, 1991 ... J.A. 606

Deposition of Bernard A. Galler, August 29, 1991 ... J.A. 624

Deposition of Bernard A. Galler, August 30, 1991 ... J.A. 629

Deposition of Ezra Gottheil, September 22, 1992 .... J.A. 632

Deposition of Frank A. Ingari, August 6, 1991 (Submitted with Dkt. No. 115) ..... J.A. 648

Deposition of Becky D. Jones, September 5, 1991 (Submitted with Dkt. No. 115) ..... J.A. 652

Deposition of Philippe Kahn, May 22, 1991 ..... J.A. 655

Deposition of David E. Kieras, October 7, 1991 ..... J.A. 663

Deposition of W. Frank King, September 12, 1991 .. J.A. 674

Deposition of Robert H. Kohn, September 11, 1991 . J.A. 678

Deposition of Thomas M. Lemberg, July 3, 1991, with Exhibit 13 annexed thereto ..... J.A. 686

Deposition of David E. Liddle, October 10, 1991 .... J.A. 692

Deposition of Joanne H. Lin, September 6, 1991 .... J.A. 706

Deposition of Jim P. Manzi in *Lotus v. Paperback*, December 30, 1987 ..... J.A. 709

PAGE

Deposition of Jim P. Manzi, August 22, 1991 .....J.A. 712

Deposition of Judith S. Olson, September 26, 1991..J.A. 717

Deposition of Judith S. Olson, October 15, 1991 ....J.A. 734

Deposition of Jack Oswald, August 29, 1991 .....J.A. 736

Deposition of John Posner in *Lotus v. Paperback*, March 7, 1989.....J.A. 743

Deposition of Jonathan M. Sachs in *Lotus v. Paperback*, November 29, 1988.....J.A. 749

Deposition of Dorothy Schrader, September 13, 1991.....J.A. 751

Deposition of Robert W. Warfield, May 15, 1991 ....J.A. 754

Deposition of Robert W. Warfield, May 16, 1991 ....J.A. 776

Deposition of Robert W. Warfield, November 9, 1992.....J.A. 784

EXHIBITS

Exhibits in Support of Lotus' Motion for Summary Judgment on Liability, filed May 7, 1991 (Declaration of Kerry L. Konrad (Dkt. Nos. 36 and 148))

Exhibit 12 .....J.A. 801

Exhibit 26 .....J.A. 809

Exhibit 30 .....J.A. 811

PAGE

Exhibit C-27 .....J.A. 814

Exhibit to Lotus' Motion for Summary Judgment, filed November 1, 1991 (Declaration of Paul M. O'Connor III (Dkt. No. 108))

Exhibit C .....J.A. 817

Exhibits in Support of Borland's Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment, dated January 30, 1992 (Dkt. No. 142)

Ex. 1: Document entitled "Quattro Pro Honors: Awards and Quotes" .....J.A. 821

Ex. 6: Letter from the Copyright Office to Kerry L. Konrad .....J.A. 827

Ex. 9: Lotus User's Manual .....J.A. 829

Ex. 14: Document entitled "Interoperability Qualitative Study" .....J.A. 831

Ex. 15: Lotus European Patent Application .....J.A. 837

Ex. 16: U.S. Patent No. 4,989,141 .....J.A. 842

Ex. 19: Letter from the Copyright Office .....J.A. 847

Ex. 20: Document from the Copyright Office entitled "Computer Programs and Related Screen Displays." .....J.A. 849

PAGE

Ex. 21: Appellants' Opening Brief in *Ashton-Tate Corp. v. Ross*, No. 89-15683 (9th Cir.).....J.A. 853

Ex. 23: U.S. Patent No. 4,611,306 .....J.A. 856

Ex. 27: Document entitled "Quattro Pro Displacement Study".....J.A. 864

Ex. 36: Text of Jim Manzi's Royal Sonesta Staff Meeting speech .....J.A. 873

Ex. 38: Document entitled "Houdini Analysis of Competitive Products".....J.A. 877

Ex. 39: Memorandum entitled "Review of Excel" ...J.A. 879

Ex. 40: Memorandum entitled "1-2-3 3.0 Positioning Study".....J.A. 881

Ex. 41: Memorandum entitled "Quattro Evaluation"..J.A. 885

Exhibit in Support of Borland International, Inc.'s Motion for Reconsideration, dated August 31, 1992 (Declaration of Andrew Konstantaras (Dkt. No. 200))

Ex. I.....J.A. 887

Documentary Exhibits Submitted at Trial, filed May 5, 1993 (Dkt. No. 359)

Ex. 7: Appendix I from Quattro Pro 3.0 User's Manual .....J.A. 899

Ex. 15: Document bearing Borland production Nos. 503897-503898 .....J.A. 907

PAGE

Ex. 18: Telegram from Philippe Kahn (Borland President) to Henry B. Gutman (counsel for Lotus) .J.A. 909

Ex. 19: Letter dated September 6, 1990 from Philippe Kahn (Borland President) to Dorothy Konrad.....J.A. 912

Ex. 32: Memorandum regarding CR4314 dated July 31, 1992.....J.A. 915

Ex. 33: Internal Bug Report .....J.A. 916

Ex. 36: Dybvig, Phillip H., *The Lotus Tutorial*.....J.A. 918

Ex. 43: Advertisement from *The Wall Street Journal*, dated August 5, 1992.....J.A. 919

Ex. 505: Advertisement from *The Wall Street Journal*, dated August 13, 1992 .....J.A.920

Ex. 507: Text of Jim Manzi's New York PC User's Group Speech .....J.A. 922

Ex. 508: Document entitled "Sales Speech" .....J.A. 923

Ex. 517: Key Reader portion of QUATTRO.MU file.J.A. 931

Exhibits From Brief of Plaintiff/Appellee from Court of Appeals for the First Circuit

Ex. A: Screen Shots (1-2-3, Quattro, Quattro Pro)...J.A. 969

Ex. B: Screen Shot (VP Planner +).....J.A. 978

PAGE

Exhibits From Briefs of Defendant/Appellant from Court of Appeals for the First Circuit

Lotus 1-2-3 Screen Display When Setting the Left Margin for Printing .....J.A. 979

Borland's Quattro Pro Screen Display in "123-Compatible Mode" When Setting the Left Margin for Printing .....J.A. 980

Lotus 123 Screen Display, Showing Available Command Choices and Long Prompt Message..J.A. 981

Borland Key Reader, with Blank Box .....J.A. 982

Microsoft's Excel Screen Display, with Lotus Menus .J.A. 983

Macintosh Screen Display .....J.A. 984

Statute

1976 Copyright Act, 17 U.S.C. § 107 .....J.A. 985

PAGE

The following items have been omitted in printing this joint appendix because they appear on the following pages in the appendix to the printed Petition for Certiorari:

Opinion of the United States Court of Appeals for the First Circuit in *Lotus Development Corporation v. Borland International, Inc.*, dated March 9, 1995. 1a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 12, 1993, as amended August 19, 1993 ... 29a

Permanent Injunction entered by United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 19, 1993 ..... 69a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated June 30, 1993, as amended August 19, 1993 ..... 71a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated July 31, 1992 ..... 106a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated March 20, 1992..... 145a

PAGE

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Paperback Software International*, dated June 28, 1990 ..... 183a

The 1976 Copyright Act, 17 U.S.C. §§ 101, 102(a), 102(b), 103, 117 ..... 270a

Affidavit of Mitchell D. Kapor, dated November 27, 1989, submitted in *Lotus Development Corporation v. Borland International, Inc.* and *Lotus Development Corporation v. Paperback Software International* 281a

RELEVANT DOCKET ENTRIES

Document Date	Docket Number[1]	Description
06/28/90	1 (N.D. Cal.)	Complaint for Declaratory Judgment, entered 06/29/90
07/02/90	1	Complaint, entered 07/03/90
07/04/90	3 (N.D. Cal.)	Amended Complaint for Declaratory Judgment, entered 07/03/90
07/11/90	5 (N.D. Cal.)	Application of Borland International, Inc. for Preliminary Injunction, entered 07/12/90
07/11/90	6 (N.D. Cal.)	Application for Temporary Restraining Order, entered 07/12/90
07/11/90	7 (N.D. Cal.)	Memorandum of Points and Authorities in Support of Motion for Temporary Restraining Order and Preliminary Injunction Enjoining Prosecution of Later Duplicate Boston Action, entered 07/12/90
07/27/90	19 (N.D. Cal.)	Notice of Cross-Motion and Cross-Motion to Dismiss, Stay or Transfer (with related Memorandum of Points and Authorities in Opposition to Motion for Preliminary Injunction and in Support of Cross-Motion to Dismiss, Stay or Transfer), entered 07/27/90
07/27/90	4	Answer and Demand for Jury Trial, entered 07/31/90

[1] Unless noted otherwise, "Docket Number" refers to the docket of the United States District Court for the District of Massachusetts, 90-CV-11662. Docket entries followed by "(N.D. Cal.)" refer to the docket in the case entitled *Borland Int'l, Inc. v. Lotus Development Corp.*, No. C 90-20386 VRW (N.D. Cal.).

Document Date	Docket Number	Description
08/03/90	23 (N.D. Cal.)	Reply Memorandum of Points and Authorities in Support of Motion for Preliminary Injunction and in Opposition to Motion to Dismiss, Stay or Transfer, entered 08/30/90
08/31/90	43 (N.D. Cal.)	Hearing before Judge Vaughn R. Walker, entered 09/05/90
09/10/90	42 (N.D. Cal.)	Order granting Lotus Development Corporation's Cross-Motion to Dismiss, Stay or Transfer for the reasons set forth in the record of the hearing held on August 31, 1990, entered 09/10/90
05/07/91	30, 147	Plaintiff's Motion for Summary Judgment on Liability, entered 05/14/91
05/07/91	32	Plaintiff's Memorandum of Law in Support of Motion for Summary Judgment on Liability, entered 05/14/91
09/30/91	88, 141	Borland's Memorandum in Support of Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment, entered 10/01/91
09/30/91	90	Videotape Exhibit to Declaration of Jack Oswald, entered 10/01/91
09/30/91	87	Borland's Motion for Summary Judgment on the Dispositive Issue of Copyrightability, entered 10/01/91

Document Date	Docket Number	Description
09/30/91	85	Brief Amicus Curiae on Behalf of the Register of Copyrights, entered 09/30/91
10/02/91	102	Brief Amicus Curiae of Copyright Law Professors, entered 10/16/91
11/01/91	120, 153	Plaintiff's Reply Memorandum in Support of Motion for Summary Judgment and in Opposition to Defendant's Cross Motion for Summary Judgment on the Issue of Copyrightability, entered 11/08/91
11/01/91	118	Videotape Exhibit to Declaration of Larry Roshfeld, entered 11/08/91
11/22/91	129, 146	Borland's Reply Memorandum in Support of Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment, entered 12/04/91
11/25/91	127	Videotape Exhibit to Declaration of Erick Hachenberg, entered 11/27/91
03/20/92	164	Opinion of the United States District Court (Keeton, J.), dated March 20, 1992 (Cert. App. 145a), entered 03/23/92
04/24/92	167	Plaintiff's Renewed Motion for Summary Judgment on Infringement, entered 04/30/92

Document Date	Docket Number	Description
04/24/92	171	Plaintiff's Memorandum of Law in Support of Renewed Motion for Summary Judgment on Infringement, entered 04/30/92
04/24/92	168	Borland's Memorandum in Support of Renewed Motion for Summary Judgment, entered 04/30/92
05/04/92	183	Borland's Response to Plaintiff's Renewed Motion for Summary Judgment, entered 05/18/92
05/04/92	177	Plaintiff's Response to Borland's Memorandum in Support of Renewed Motion for Summary Judgment, entered 05/7/92
06/29/92	189	Borland's Motion for Leave to Submit Supplemental Memorandum re: Additional Authority, entered 06/29/92
07/14/92	192	Lotus' Response to Borland's Supplemental Memorandum re: Additional Authority, entered 07/15/92
07/31/92	193	Opinion of the United States District Court (Keeton, J.), dated July 31, 1993 (Cert. App. 106a), entered 08/03/92
08/31/92	201	Borland International's Motion for Certification, entered 09/01/92
08/31/92	202	Memorandum of Points and Authorities in Support of Borland's Motion to Certify, entered 09/01/92

Document Date	Docket Number	Description
08/31/92	197	Borland's Motion for Reconsideration of Order dated July 31, 1992, entered 09/01/92
08/31/92	198	Memorandum of Points and Authorities in Support of Borland's Motion for Reconsideration, entered 09/01/92
09/14/92	212	Plaintiff's Memorandum of Law in Opposition to Borland's Motion for Reconsideration, entered 09/16/92
09/14/92	210	Plaintiff's Memorandum of Law in Opposition to Borland International's Motion for Certification, entered 09/16/92
09/18/92	227	Reply Memorandum in Support of Borland's Motion to Certify, entered 09/23/92
09/18/92	246	Reply Memorandum in Support of Borland's Motion for Reconsideration, entered 11/30/92
09/23/92	241	Conference before Judge Keeton, denying Motion to Certify Question re: Copyrightability for Interlocutory Appeal, entered 10/02/92
10/15/92	239	Brief of Amicus Curiae Concerning Certification of Question for Interlocutory Appeal, entered 10/16/92
10/16/92	242	Hearing before Judge Keeton, denying Defendant's Motion for

Document Date	Docket Number	Description
		Reconsideration of the Order dated July 31, 1992, and denying Defendant's request for reconsideration of denial of Defendant's Motion to Certify Question for Interlocutory Appeal, entered 10/22/92
12/07/92	250	Plaintiff's Motion for Leave to File a Supplemental Complaint, entered 12/08/92
12/07/92	257	Plaintiff's Memorandum of Law in Support of Motion for Leave to File a Supplemental Complaint, entered 12/17/92
12/21/92	258	Borland's Memorandum of Law in Opposition to Lotus' Motion for Leave to Supplement Complaint, entered 12/21/92
01/14/93	310	Hearing before Judge Keeton, granting Motion for Leave to File Supplemental Complaint; parties waive jury trial, entered 01/27/93
01/19/93	305	Supplemental Complaint, entered 01/20/93
01/21/93	307	Answer to Supplemental Complaint, entered 01/21/93
01/29/93	320	Telephone Conference before Judge Keeton re: upcoming trial, entered 02/02/93
02/01/93	351	Day One of Nonjury Trial (Phase I), entered 03/24/93

Document Date	Docket Number	Description
02/02/93	332	Day Two of Nonjury Trial (Phase I), entered 02/18/93
02/03/93	330	Stipulation and Order Regulating Trial, entered 02/10/93
02/03/93	333	Day Three of Nonjury Trial (Phase I), entered 02/18/93
02/19/93	338	Plaintiff's Key Reader Trial Memorandum, entered 02/25/93
02/26/93	341	Borland's Motion to Amend Original Answer (with Amended Answer and Demand for Jury Trial), entered 03/01/93
02/26/93	344	Plaintiff's Fair Use Memorandum, entered 03/04/93
03/08/93	345	Borland's Response to Plaintiff's Key Reader Trial Memorandum, entered 03/10/93
03/11/93	346	Borland's Response to Plaintiff's Fair Use Memorandum, entered 03/15/93
03/12/93	347	Plaintiff's Memorandum in Opposition to Motion to Amend Original Answer, entered 03/15/93
03/30/93	353	Memorandum and Order (allowing Defendant's Motion to Amend Answer and denying Defendant's request for judgment), entered 03/30/93
03/31/93	403	Day Four of Nonjury Trial (Phase II), entered 08/31/93

Document Date	Docket Number	Description
04/01/93	404	Day Five of Nonjury Trial (Phase II), entered 08/31/93
04/02/93	405	Day Six of Nonjury Trial (Phase II), entered 08/31/93
05/05/93	361	Computer For Use With Software Exhibits filed by the Parties, entered 05/05/93
06/30/93	374	Opinion of the United States District Court dated June 30, 1993 (Keeton, J.), as amended August 19, 1993 (Cert. App. 71a), entered 07/01/93
07/27/93	378	Borland's Renewed Motion for Certification, entered 07/29/93
07/27/93	379	Memorandum of Points and Authorities in Support of Borland's Renewed Motion to Certify, entered 07/29/93
08/10/93	388	Plaintiff's Memorandum of Law in Opposition to Borland's Renewed Motion for Certification, entered 08/13/93
08/12/93	398	Opinion of the United States District Court dated August 12, 1993 (Keeton, J.), as amended August 19, 1993 (Cert. App. 29a), entered 08/24/93
08/16/93	390	Reply Memorandum in Support of Borland's Renewed Motion to Certify, entered 08/24/93
08/17/93	395	Brief of Amici Curiae User Groups In Support of Certification of

Document Date	Docket Number	Description
		Orders for Appeal, entered 08/24/93
08/19/93	406	Hearing before Judge Keeton re: Renewed Motion to Certify, entry of Permanent Injunction, and amendment of prior opinions re: confidential information, entered 08/31/93
08/19/93	397	Permanent Injunction (Cert. App. 69a), entered 08/24/93
09/10/93	408	Notice of Appeal of Defendant Borland International, entered 09/14/93
12/03/93	(1st Cir.)	Brief Amicus Curiae of Copyright Law Professors, entered 12/23/93
12/08/93	(1st Cir.)	Brief of Amicus Curiae of Software Entrepreneurs' Forum, entered 12/23/93
12/08/93	(1st Cir.)	Brief Amicus Curiae of Computer Software Industry Association, entered 12/23/93
12/10/93	(1st Cir.)	Brief Amicus Curiae of American Committee for Interoperable Systems, entered 12/23/93
12/14/93	(1st Cir.)	Brief of Defendant/Appellant Borland International, Inc., entered 12/14/93
12/14/93	(1st Cir.)	Brief Amicus Curiae of Professor Dennis S. Karjala and Professor Peter S. Menell, entered 12/23/93

Document Date	Docket Number	Description
12/14/93	(1st Cir.)	Amicus Brief of Computer Scientists Re Copyrightability of Computer Languages, entered 12/23/93
12/14/93	(1st Cir.)	Brief of Amici Curiae (Chicago Computer Society, Diablo Users Group, Danbury Area Computer Society, IBM Ab Users Group, Kentucky-Indiana Personal Computer Users Group, Long Island PC Users Group, Napa Valley PC Users Group, Pacific Northwest PC Users Group, Palmetto Personal Computer Club, Philadelphia Area Computer Society, Inc., Phoenix IBM PC Users Group, Pinellas IBM PC Users Group, Quad Cities Computer Society, Inc., Quattro Pro Users Group, Sacramento PC Users Group, San Francisco PC Users Group, Santa Barbara Twin Cities PC Users Group, Warner Robbins Personal Computer Association), entered 12/23/93
01/27/94	(1st Cir.)	Brief of Amicus Curiae Computer and Business Equipment Manufacturers Association, entered 02/10/94
01/28/94	(1st Cir.)	Brief of Amici Curiae Adobe Systems Inc.; Apple Computer Inc.; Computer Associates International Inc.; Digital Equipment Corporation; and International Business

Document Date	Docket Number	Description
		Machines Corporation, entered 02/10/94
01/28/94	(1st Cir.)	Brief of Amici Curiae Apple Computer Inc., Digital Equipment Corporation, International Business Machines Corporation, and Xerox Corporation, entered 02/10/94
01/28/94	(1st Cir.)	Brief of Plaintiff/Appellee Lotus Development Corporation, entered 01/31/94
02/14/94	(1st Cir.)	Reply Brief of Defendant/Appellant Borland International, Inc., entered 02/18/94
03/01/94	(1st Cir.)	Motion for Leave to Add Signatures to Amicus Brief of Computer Scientists, entered 03/01/94
03/01/94	(1st Cir.)	Order Granting Motion for Leave to Add Signatures to Amicus Brief of Computer Scientists, entered 03/01/94
04/06/94	(1st Cir.)	Oral Argument (Breyer, C.J., Torruella, Stahl, JJ.), entered 04/06/94
07/18/94	(1st Cir.)	Order: upon recusal of Chief Judge Breyer, case rescheduled for rehearing at the October 1994 Session, entered 07/18/94
10/06/94	(1st Cir.)	Oral Argument (Torruella, Boudin, Stahl, JJ.), entered 10/06/94

Document Date	Docket Number	Description
03/09/95	(1st Cir.)	Opinion of the United States Court of Appeals for the First Circuit (Cert. App. 1a), entered 03/09/95
03/31/95	678	Informal Mandate of Court of Appeals, entered 03/31/95

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

COMPLAINT

Plaintiff, LOTUS DEVELOPMENT CORPORATION, by its undersigned attorneys, complains of the defendant and alleges as follows:

NATURE OF THE ACTION

1. This is an action for copyright infringement arising out of defendant's sale of computer software programs called "Quattro" and "Quattro Pro". Both programs contain deliberate, command-for-command copies of the user interface, and in particular of the menu structure and commands, of plaintiff's highly successful and copyrighted computer software program, "Lotus 1-2-3".

JURISDICTION AND VENUE

2. This Court has jurisdiction over the subject matter of this action pursuant to 28 U.S.C. Sections 1331, 1332 and 1338.

3. Venue is proper in this Court pursuant to 28 U.S.C. Sections 1391 and 1400, in that defendant is and was at all relevant times doing business in this judicial district on its own behalf and through its agents, and may be found in this judicial district, and the claims asserted herein arise, at least in part, in this judicial district.

THE PARTIES

4. Plaintiff LOTUS DEVELOPMENT CORPORATION ("Lotus") is a corporation organized and existing under the laws of the State of Delaware, with its principal place of business at 55 Cambridge Parkway, Cambridge, Massachusetts. Lotus is principally engaged in the business of designing, developing and marketing computer application software programs. "1-2-3" is a registered trademark for the software products sold under that name by plaintiff. "Lotus" is a registered trademark of Lotus Development Corporation.

5. Defendant BORLAND INTERNATIONAL, INC. ("Borland"), upon information and belief, is a corporation organized and existing under the laws of the State of California with its principal place of business at 1800 Green Hills Road, Scotts Valley, Santa Cruz County, California. Upon information and belief, Borland is engaged in the business of marketing and distributing personal computer programming language and application software programs on a nationwide basis, including within the District of Massachusetts, on its own behalf and through its agents.

THE COPYRIGHTED WORKS

6. In 1982, employees of Lotus created and developed an original computer software program entitled "Lotus 1-2-3" or "1-2-3". Lotus 1-2-3 provides users of certain computers with the ability to perform certain tasks with their computers, most notably the accumulation, arrangement and calculation of data and text in what is known as a "spreadsheet" format. Lotus 1-2-3 also contains database storage and management and graphics presentation capabilities which are connected or "integrated" with its spreadsheet capabilities. Lotus 1-2-3 is sold as part of a package which includes associated user instruction manuals and other documentation.

7. Lotus 1-2-3 communicates with users of the program through its "user interface". The user interface, *inter alia*, displays a series of menus arranged in a hierarchical "menu structure", which present the user with a variety of choices for operating the program, called "commands" or "menu commands".

8. Lotus 1-2-3, including its user interface and menu structure, contains material wholly original with and developed by Lotus and is copyrightable subject matter under the laws of the United States.

9. The original version of Lotus 1-2-3, called "Lotus 1-2-3, Release 1", was first sold to the public by Lotus in January 1983.

10. Certain revisions were made to Lotus 1-2-3, Release 1, and the revised edition of the program was sold to the public as "Lotus 1-2-3, Release 1A", commencing in mid-1983. Release 1A, including its user interface and menu structure, is substantially similar to and derived from Release 1.

11. Commencing in September 1985, Lotus published and began to ship a revised version of Lotus 1-2-3, known as "Release 2.0". Release 2.0, including its user interface and menu structure, is substantially similar to and derived from

the original version of Lotus 1-2-3. As used herein, "Lotus 1-2-3" refers to Release 1, Release 1A and Release 2.0.

12. Lotus complied in all respects with the provisions of 17 U.S.C. Section 101 *et seq.* and all other laws governing copyright, to secure the exclusive rights and privileges in and to copyright for Lotus 1-2-3. Lotus received from the Registrar of Copyrights certificates of registration for Lotus 1-2-3 as follows:

Title	Certificate NO.	Date
Lotus 1-2-3 (Release 1)	TX 1-223-501	12/23/83
Lotus 1-2-3 (Release 1A)	TX 1-223-502	12/23/83
Lotus 1-2-3 (Release 2.0)	TX 1-694-602	11/08/85

13. Each and every element of original and copyrightable subject matter contained in Lotus 1-2-3, including its user interface and menu structure, and each and every element of copyrightable subject matter which Lotus alleges defendant has infringed herein, is and has at all times been fully covered by its registered copyrights for Lotus 1-2-3.

14. Since its inception, Lotus 1-2-3 has been legally produced only by Lotus or by its authorized licensees and all copies of it made by Lotus have been offered with notice of copyright affixed therein and on the package in strict conformity with the provisions of the federal copyright laws.

15. At all times, Lotus has been and is the sole proprietor of all rights, title and interest in and to the copyrights in Lotus 1-2-3.

THE ACTS OF INFRINGEMENT

16. Since November 1987 or earlier, the exact date as yet being unknown to plaintiff, defendant infringed plaintiff's copyrights in Lotus 1-2-3 by copying, publishing, distributing, marketing and placing upon the public market in interstate commerce, and/or causing the copying, publishing, distributing, marketing and placing in interstate commerce of, a computer software program called "Quattro".

17. Quattro contains, as one of two user interfaces provided to users of the program, a user interface copied from, and substantially similar or identical to, Lotus 1-2-3 in numerous respects, including, in particular, its menu structure and command choices. Attached hereto as Exhibit A is a true copy of a chart included in the Quattro "Reference Guide", sold together with the Quattro program, which depicts the "Menu Tree" or menu structure of the infringing user interface. As the chart indicates, only those menu items indicated with a square bullet are not found in Lotus 1-2-3; such items are described in the chart as "extensions to Lotus 1-2-3 menus".

18. Commencing in November 1989 or earlier, the exact date being unknown to plaintiff, defendant has infringed plaintiff's copyrights in Lotus 1-2-3 by copying, publishing, distributing, marketing and placing in interstate commerce, and/or causing the copying, publishing, distributing, marketing and placing in interstate commerce of, a computer software program called "Quattro Pro".

19. Quattro Pro contains, as one of three user interfaces provided to users of the program, a user interface copied from, and substantially similar or identical to, Lotus 1-2-3 in numerous respects, including, in particular, its menu structure and command choices.

20. Upon information and belief, Quattro Pro is derived in substantial part from Quattro, including those elements that are copied from Lotus 1-2-3.

21. The deliberate aim and result of defendant's copying is to provide a "clone" or identical recreation of substantial and significant elements of the Lotus 1-2-3 user interface, and in particular of its menu structure, in defendant's Quattro and Quattro Pro programs.

22. Indeed, the Quattro Reference Guide specifically refers (at page 22) to the menu structure copied from Lotus 1-2 3 as the "Lotus-compatible menu tree". The "User's Guide" to Quattro Pro, sold together with that program, refers to the

menu structure copied from Lotus 1-2-3 as the "123-compatible menu tree" (at page 4) and as the "123-compatible menus" (passim). Virtually every command contained in the Lotus 1-2-3 menus is not only replicated in the Quattro and Quattro Pro "compatible menus", but also is presented in virtually the same structure, sequence and organization as in Lotus 1-2-3, and instructs the computer to perform the same tasks.

23. A user of Quattro or Quattro Pro can invoke the menu structure copied from Lotus 1-2-3 by typing "Q123" when initially loading either program into the computer's resident memory. Alternatively, each program contains a menu command called "Menu Tree" which provides the user with the ability to cause either program to display the copied Lotus 1-2-3 menu structure. Following invocation of the "Menu Tree" command, selection of the 1-2-3 menu structure is indicated to the Quattro user by the preset notations "123.RSC" or "123.ALT"; in Quattro Pro by the preset notation "123.MU".

24. Borland's acts were performed without the permission, license or consent of Lotus.

25. As a result of defendant's infringement of plaintiff's copyrights in Lotus 1-2-3, plaintiff has suffered, and is continuing to suffer, irreparable injury. The copyrights infringed by defendant contribute fundamentally to plaintiff's revenues, commercial prestige, goodwill and profits. Defendant has sold and/or disseminated or caused to be sold and/or disseminated, and intends to sell and/or disseminate or cause to be sold and/or disseminated, its infringing products to plaintiff's clients, customers, and others, and to otherwise use or cause to be used its infringing products for its own commercial purposes. In so doing, defendant has caused and will cause Lotus to lose current and potential business, market position, goodwill, prestige and profits all related to the sale of Lotus 1-2-3. Plaintiff cannot be adequately compensated for these actual and threatened injuries by damages alone.

26. By reason of defendant's acts of infringement alleged above, Lotus has suffered damage to its business in the form of diversion of trade, loss of profits, injury to goodwill and reputation, and a dilution in the value of its rights, all in amounts which are not yet fully ascertainable.

27. Defendant's acts of infringement alleged above were performed willfully, intentionally and deliberately. Defendant commenced sale of Quattro and Quattro Pro after Lotus had publicly declared its belief that the user interface and menu structure of Lotus 1-2-3 were covered by its copyrights, and after Lotus had expressed in word and deed its determination to enforce its legal rights thereto. Defendant had full knowledge of the various expressions by Lotus of its legal position commencing as early as January 1987.

28. Because defendant's acts of infringement alleged above were willful, intentional and deliberate, Lotus is entitled to an award of its attorney's fees incurred in this action pursuant to 17 U.S.C. Section 505.

29. Defendant's willful and deliberate acts of infringement entitle Lotus to an award of its actual damages incurred and/or to an accounting by the defendant of any profits, gains and advantages derived by it as a result of the infringing activities, or to an award of the maximum measure of statutory damages for each act of infringement.

WHEREFORE, plaintiff prays for judgment as follows:

1. Under the authority of 17 U.S.C. Section 502, for a preliminary and permanent injunction restraining Borland from, *inter alia*, distributing, publishing or marketing the computer software programs called Quattro or Quattro Pro or in any way infringing Lotus' copyrights in Lotus 1-2-3;

2. Under the authority of 17 U.S.C. Section 1118, for an Order requiring the impounding and destruction of all Borland's products or goods that infringe any of plaintiff's copyrights;

3. For an Order awarding Lotus actual damages incurred as the result of Borland's acts of infringement, and/or an accounting to Lotus by Borland of any profits derived by it from its infringing activities; or in the alternative awarding Lotus the maximum statutory damages for each act of infringement committed by Borland;

4. Under the authority of 17 U.S.C. Section 505, for an Order awarding Lotus its attorney's fees incurred in this action;

5. For costs of suit incurred in this action; and

6. For such other and further relief as this Court may deem proper.

Dated: July 2, 1990

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman
30 Rockefeller Plaza
New York, New York 10112
(212) 408-2400

HALE AND DORR

By: JEFFREY B. RUDMAN
JAMES C. BURLING
Jeffrey B. Rudman
James C. Burling
60 State Street
Boston, Massachusetts 02109
(617) 742-9100

Attorneys for Plaintiff
Lotus Development Corporation

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

v.

BORLAND INTERNATIONAL, INC.,

Defendant.

ANSWER AND DEMAND FOR JURY TRIAL

Defendant Borland International, Inc. ("Borland") answers the complaint of Lotus Development Corporation ("Lotus") as follows:

FIRST DEFENSE

1. Borland admits that plaintiff purports to bring an action for copyright infringement arising out of Borland's sale of computer software programs called "Quattro" and "Quattro Pro". Except as expressly admitted herein, Borland denies each and every allegation contained in paragraph 1.

2. Borland admits that the Court has subject matter jurisdiction as conferred by 28 U.S.C. §§1331 and 1338. Borland denies each and every other allegation contained in paragraph 2.

3. Borland admits that it is doing business in this judicial district. Borland denies that venue is proper in this Court.

Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 3.

4. Borland admits that Lotus is a corporation organized and existing under the laws of the State of Delaware, with its principal place of business in Cambridge, Massachusetts. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 4.

5. Borland admits that its principal place of business is at 1800 Green Hills Road, Scotts Valley, Santa Cruz County, California. Borland further admits that it is engaged in the business of marketing and distributing personal computer programming language and application software programs on a nationwide basis, including within the District of Massachusetts. Borland denies the remaining allegations contained in paragraph 5.

6. Borland admits that there is a computer software program entitled "Lotus 1-2-3"; that Lotus 1-2-3 provides users of certain computers with the ability to perform certain computer-assisted tasks including the accumulation, arrangement, and calculation of data and text in what is known as a spreadsheet format; that Lotus 1-2-3 also contains certain database storage and management and graphics presentation capabilities; and that Lotus 1-2-3 is sold as part of a package which includes user instruction manuals and other documentation. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 6.

7. Borland admits that users of Lotus 1-2-3 communicate with Lotus 1-2-3 by various means. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 7.

8. Borland denies that Lotus 1-2-3 is copyrightable subject matter under the laws of the United States, at least to the

extent apparently claimed by Lotus for the purposes of the present action. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 8.

9-15. Borland is without knowledge or information sufficient to form a belief as to the truth of the allegations within paragraphs 9, 10, 11, 12, 13, 14, and 15.

16. Borland denies each and every allegation contained in paragraph 16.

17. Borland admits that Quattro offers users the capability to create an unlimited number of alternative commands and offers several examples of command alternatives, and that Exhibit A to the Complaint is a copy of a chart that at certain times has been included in the Quattro "Reference Guide" provided with the Quattro program. Borland denies each and every other allegation contained in paragraph 17.

18-26. Borland denies each and every allegation contained in paragraphs 18, 19, 20, 21, 22, 23, 24, 25, and 26.

27. Borland admits that Lotus has made various statements regarding its copyright positions at various times. Borland denies each and every other allegation contained in paragraph 27.

28-29. Borland denies each and every allegation contained in paragraphs 28 and 29.

SECOND DEFENSE

Lotus has failed to state a claim upon which relief can be granted.

THIRD DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu structure and commands as alleged in the Complaint, at

least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a valid copyright therein, because the Lotus 1-2-3 menu structure and commands constitute uncopyrightable subject matter.

FOURTH DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu structure and commands as alleged in the Complaint, at least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a copyright therein, because such features lack originality.

FIFTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of laches. Borland has manufactured, used, and sold its Quattro products for a substantial period of time. Despite its knowledge of such activities, Lotus failed to assert the claims raised in the Complaint within a reasonable period of time. Borland relied upon this failure by Lotus and would be prejudiced were Lotus permitted now belatedly to assert the claims contained in its Complaint.

SIXTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of estoppel.

SEVENTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of waiver.

EIGHTH DEFENSE

Borland has not copied, published, distributed, marketed, or placed in interstate commerce any computer software program which infringes upon or violates any rights of Lotus, including any rights that might exist under copyright.

NINTH DEFENSE

This Court should dismiss or stay this case on the ground that a case involving the same parties and addressing the identical issues was filed by Borland before Lotus commenced this action and is pending in the United States District Court for the Northern District of California.

TENTH DEFENSE

For the convenience of the parties and witnesses, in the interest of justice, this Court should transfer this action to the United States District Court for the Northern District of California.

WHEREFORE, defendant Borland International, Inc. prays for judgment as follows:

A. That the relief requested by Lotus be denied and that the Complaint be dismissed with costs and attorneys' fees to defendant;

B. That such other and further relief be granted as the Court deems just and proper.

JURY DEMAND

Pursuant to Fed. R. Civ. P. 38(b), defendant hereby demands a trial by jury of all issues so triable.

July 27, 1990 /s/ LAURA STEINBERG

Laura Steinberg
BBO #478440
SULLIVAN & WORCESTER
One Post Office Square
Boston, MA 02109
(617/338-2867)

Attorney for defendant
Borland International, Inc.

CERTIFICATE OF SERVICE

I hereby certify that on this day a true copy of the above document was served upon the attorney of record for each party by hand & by federal express.

Date 7/27/90 Laura Steinberg

UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORP.,

Plaintiff,

v.

BORLAND INTERNATIONAL,

Defendant.

BRIEF AMICUS CURIAE ON BEHALF OF THE REGISTER OF COPYRIGHTS

STUART M. GERSON
Assistant Attorney General

WAYNE BUDD
United States Attorney

PAUL G. LEVENSON
Assistant United States Attorney
1107 John W. McCormack Federal Bldg.
U.S. Post Office & Courthouse
Boston, Massachusetts 02109
Tel: (617) 223-9400

VITO J. DIPIETRO
Director
Commercial Litigation Branch

JOHN FARGO
Assistant Director
Commercial Litigation Branch
Civil Division
Department of Justice
Washington, D.C. 20530
Tel: (202) 307-0458

Of Counsel:

RALPH OMAN
DOROTHY SCHRADER
CHARLOTTE DOUGLASS
Attorneys
U.S. Copyright Office

DRAFT FOR INTERNAL USE ONLY
September 29, 1989

The following practices are derived from actual screen display cases examined under the policy established in June 1988. They are not to be considered final, but evolutionary. The Examining Division reserves the right to examine new cases on their own merits.

SCREEN DISPLAY PRACTICES

A computer program and its related copyrightable screen displays, when owned by the same claimant, must be registered together in a single registration. Generally, when an applicant registers a computer program, no subsequent registration will be permitted for the screens associated with that program. The applicant has the option of sending the required i.d. material for the related screen displays with the i.d. material for the computer program, and has the option of stating a claim in the screen displays on the application.

If a specific claim is made in the screen displays, i.d. material must be deposited for them. All i.d. material for screen displays, even if not expressly claimed on the application, will be examined for copyrightability. If the screens are not copyrightable, the i.d. material for them will be removed from the deposit and retained in a correspondence file. The Office will not allow a specific reference to screen displays if the screens are found not to be copyrightable.

Once registration is made for the computer program, it is considered to extend to any copyrightable screen displays embodied in the program. No second registration will be made for the same version of the work in order to put a specific claim in screen displays on the record. Likewise, if an applicant chooses to register only the screen displays, the

Office will not make a later registration for the computer program.

* * *

[6] 5. Nature of authorship statement includes "menu screens".

a. If the menu screen contains copyrightable authorship *aside from* the menu listings, ask for better statement of authorship, such as "text of screen displays" or "artwork in screen displays".

b. If there is no copyrightable authorship, reject menu screens, which, like a table of contents, are functionally determined and are generally not registrable.

6. Nature of authorship described as "audiovisual display of screens" or "visual displays" or space 1 may refer to "motion picture" or "audiovisual work."

a. Videotape for textual screens and i.d. material for computer program deposited.

Request appropriate i.d. material for textual screens and a Form TX with specific authorship statement. (See item 5 under DEPOSIT.)

b. Videotape for pictorial (not audiovisual) screens and i.d. material for computer program deposited.

Request appropriate i.d. material for screens and Form TX or VA with specific authorship statement, e.g. "artwork on screens, text of computer program," or "technical drawings and text of computer program."

c. Videotape containing only audiovisual material deposited.

Offer computer program registration and request i.d. material for the computer program. Request new application describing authorship as "entire computer program including audiovisual material in screen displays".

* * *

[12] 4. Screen consists of a form designed for recording information that the Office would register as a compilation if fixed in another medium.

Register. Screens are judged by traditional standards of copyrightability. (Be sure it is not a menu-type screen or other screen that may be dictated by function. These are not registrable.)

NOTE: Nature of authorship must state "compilation of terms in screen display(s)". Consult with a Literary Section Head in all these cases.

5. Icons of commonly used symbols, such as a trash can, pencil or file drawer claimed.

Apply de minimis/trivial standards carefully, but do not suggest that the symbol may have become "standard" with a large body of the public.

NOTE: If the claim is in "iconic programming" or "object-oriented programming", such as programming in the Double-Helix language, be sure that space 6 disclaims the icons, which are part of the underlying program. (These are not screen display claims, though the deposit for these computer programs looks like a set of screen displays.)

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

SUPPLEMENTAL COMPLAINT

Pursuant to Rule 15(d), Fed. R. Civ. P., plaintiff LOTUS DEVELOPMENT CORPORATION ("Lotus"), by its undersigned attorneys, hereby supplements its original complaint in this action as follows:

COUNT II
(*Additional Acts of Copyright Infringement*)

1. Since November 1990 or earlier, the exact date as yet being unknown to plaintiff, defendant Borland International, Inc. ("Borland") has infringed plaintiff's copyrights in Lotus 1-2-3, including plaintiff's exclusive right to prepare works derived from the copyrighted work, by copying, publishing, distributing, marketing and placing in interstate commerce and/or causing the copying, publishing, distributing, marketing and placing in interstate commerce, of computer software programs containing a feature described as the "Macro Key Reader" or "Key Reader". These programs are included

in products sold by Borland under the names "Quattro Pro 2.0", "Quattro Pro 3.0", "Quattro Pro 4.0", "Quattro Pro SE", "Quattro Pro 4.01" and "Quattro Pro for Windows".

2. In implementing the Key Reader, Borland has employed a deliberate copy of, and infringing work derived from, the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences.

3. Borland first published Quattro Pro 4.01 and Quattro Pro for Windows subsequent to this Court's determination in this action that the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences constitute copyrightable expression and are protected by Lotus' copyrights, and with full knowledge of that decision.

4. All of the acts alleged above were performed by Borland without the permission, license or consent of Lotus.

5. Borland's acts of infringement alleged above are willful, intentional and deliberate.

6. As a result of defendant's infringement of plaintiff's copyrights in Lotus 1-2-3, including plaintiff's exclusive right to prepare works derived from the copyrighted work, plaintiff has suffered, and is continuing to suffer, irreparable injury. The copyrights infringed by defendant, including the exclusive right to prepare works derived from the copyrighted work, contribute fundamentally to plaintiff's revenues, commercial prestige, goodwill and profits. Defendant has sold and/or disseminated or caused to be sold and/or disseminated, and intends to sell and/or disseminate or cause to be sold and/or disseminated, its infringing products to plaintiff's clients, customers, and others, and to otherwise use or cause to be used its infringing products for its own commercial purposes. In so doing, defendant has caused and will continue to cause plaintiff to lose current and potential business, market position, goodwill, prestige and profits related to the sale of Lotus 1-2-3. Plaintiff cannot be adequately compensated for these actual and threatened injuries by damages alone.

7. By reason of defendant's acts of infringement alleged above, Lotus has suffered damage to its business in the form of diversion of trade, loss of profits, injury to goodwill and reputation and a dilution in the value of its rights, including the exclusive right to prepare works derived from the copyrighted work, all in amounts which are not yet fully ascertainable.

WHEREFORE, plaintiff prays for judgment as follows:

1. Under the authority of 17 U.S.C. Section 502, for a preliminary and permanent injunction restraining Borland from, *inter alia*, distributing, publishing or marketing the computer software products called Quattro Pro 2.0, Quattro Pro 3.0, Quattro Pro 4.0, Quattro Pro SE, Quattro Pro 4.01 and Quattro Pro for Windows, or any other product containing computer software programs employing Borland's current implementation of the Key Reader, or in any way infringing Lotus' copyrights in Lotus 1-2-3, including plaintiff's exclusive right to prepare works derived from the copyrighted work.

2. Under the authority of 17 U.S.C. Section 518, for an Order requiring the impounding and destruction of all of defendant's products or goods that infringe any of plaintiff's copyrights, including plaintiff's exclusive right to prepare works derived from the copyrighted work.

3. For an Order awarding plaintiff actual damages incurred as the result of defendant's acts of infringement and/or an accounting to plaintiff by defendant of any profits, gains and advantages derived by it from its infringing activities, or in the alternative awarding plaintiff the maximum statutory damages for each act of infringement committed by defendant;

4. Under the authority of 17 U.S.C. Section 505, for an Order awarding plaintiff its attorney's fees incurred in this action;

5. For costs of suit incurred in this action; and

6. For such other and further relief as this Court may deem proper.

Dated: January 14, 1993

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman
(admitted *pro hac vice*)

30 Rockefeller Plaza
New York, New York 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, Massachusetts 02109
(617) 742-9100

Attorneys for Plaintiff
Lotus Development Corporation

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

ANSWER TO SUPPLEMENTAL COMPLAINT

Defendant Borland International, Inc. ("Borland") answers the Supplemental Complaint of plaintiff Lotus Development Corporation ("Lotus") as follows:

1. Borland admits that, beginning in late 1990, Borland has sold and distributed products marketed as "Quattro Pro 2.0", "Quattro Pro 3.0", "Quattro Pro 4.0", "Quattro Pro SE", "Quattro Pro 4.01" and "Quattro Pro for Windows", each of which contains a feature known as the "Macro Key Reader" or "Key Reader". Except as expressly admitted herein, Borland denies each and every allegation in paragraph 1.

2. Borland denies each and every allegations contained in paragraph 2.

3. Borland admits that Quattro Pro 4.01 and Quattro Pro for Windows were both first sold and distributed following this Court's July 31, 1992 decision in this action. Except as expressly admitted herein, Borland denies each and every allegation contained in paragraph 3.

4. Borland denies each and every allegation contained in paragraph 4.

5. Borland denies each and every allegation contained in paragraph 5.

6. Borland denies each and every allegation contained in paragraph 6.

7. Borland denies each and every allegation contained in paragraph 7.

FIRST AFFIRMATIVE DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences as alleged in the Supplemental Complaint, at least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a valid copyright therein, because the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences constitute uncopyrightable subject matter.

SECOND AFFIRMATIVE DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences as alleged in the Supplemental Complaint, at least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a copyright therein, because such features lack originality.

THIRD AFFIRMATIVE DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of laches, particularly with respect to the Key Reader. Bor-

land has manufactured, used, and sold the Key Reader feature in its Quattro Pro products for a substantial period of time. Despite its knowledge of such activities, Lotus failed to assert the claims raised in the Complaint within a reasonable period of time. Borland relied upon this failure by Lotus and would be prejudiced were Lotus now permitted to belatedly assert the claims contained in its Supplemental Complaint.

FOURTH AFFIRMATIVE DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of estoppel, particularly with respect to the Key Reader.

FIFTH AFFIRMATIVE DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of waiver, particularly with respect to the Key Reader.

SIXTH AFFIRMATIVE DEFENSE

To the extent, if any, Lotus has a valid copyright in the Lotus 1-2-3 menu command hierarchy, macro language and keystroke sequences as alleged in the Supplemental Complaint, Borland's use thereof constitutes a fair use as allowed for in 17 U.S.C. §107.

WHEREFORE, defendant Borland International, Inc. prays for judgment as follows:

A. That the relief requested by Lotus be denied and that the Supplemental Complaint be dismissed with costs and attorneys' fees to defendant;

B. That such other and further relief be granted as the Court deems just and proper.

Dated: January __, 1993

By: GARY L. REBACK
Gary L. Reback

WILSON, SONSINI, GOODRICH & ROSATI
Professional Corporation
Two Palo Alto Square
Palo Alto, California 94306
Telephone: (415) 493-9300

Peter E. Gelhaar
DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, Massachusetts 02110
Telephone: (617) 261-9890

Attorneys for Defendant BORLAND INTERNATIONAL, INC.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

FILED IN OPEN COURT
Date: 2/3/93
Lowrie Deputy Clerk

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

STIPULATION AND ORDER REGULATING TRIAL

Plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") hereby agree as follows:

1. *Scope*. This Stipulation and Order is to govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial"). The parties' waivers of their respective rights to trial by jury are as set forth on page 63 of the January 14, 1993 hearing transcript.

2. The parties have entered into a stipulation concerning the "long prompts" in the parties' respective products (a copy of which is annexed hereto as Exhibit A), the terms of which are incorporated herein by reference.

3. The parties hereby consent to the introduction of designated affidavits, declarations or deposition testimony from this action or *Lotus Development Corp. v. Paperback Software International*, 87-0076-K (D.Mass.), at the Phase I Trial, in lieu of oral, direct examination, as if it were given live in Court herein.

4. The parties expressly waive any right to adverse direct examination, cross-examination or curative direct examination in Court of any witness, including those whose testimony is introduced by either party pursuant to the preceding paragraph. The parties have made their final designation of testimony and other proof, unless otherwise ordered by the Court.

5. The parties agree to withdraw any objection to the introduction of evidence on the ground of relevance that is based upon an asserted violation of the Order Regulating Jury Trial, entered September 30, 1992, but reserve the right to argue that any such evidence should be given no weight by the Court as trier of fact in reaching its determinations herein.

6. Each party agrees not to assert any objection to the other party's counter-designations of deposition testimony pursuant to Rule 32(a)(4), Fed. R. Civ. P., on the ground that the counter-designation is unnecessary to make the designated excerpt complete and not misleading.

7. The parties agree to a schedule for resolving any issue raised by Lotus' Supplemental Complaint as set forth in the agreement annexed hereto as Exhibit B. The parties further agree to attempt in good faith to agree upon a procedure for the resolution of disputed facts, if any, pertinent to determination of such issues, including, but not limited to, a procedure comparable to that set forth herein.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development
Corporation

SO ORDERED:

ROBERT E. KEETON
Robert E. Keeton, U.S.D.J.

Date: 2/3/93

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

FILED IN OPEN COURT
Date: 2/2/93
Lowrie Deputy Clerk

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

STIPULATION AND ORDER REGULATING TRIAL

Plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") hereby agree as follows:

1. *Scope.* This Stipulation and Order is to govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial"). The parties, ~~have waived any~~ [waiver of their respective] right[s] to trial by jury ~~of all issues herein other than damages.~~ [are as set forth on page 63 of the January 14, 1993 hearing transcript.]

2. The parties have entered into a stipulation concerning the "long prompts" in the parties' respective products (a copy of which is annexed hereto as Exhibit A), the terms of which are incorporated herein by reference.

3. The parties hereby consent to the introduction of [designated] affidavits, declarations or deposition testimony from this action or *Lotus Development Corp. v. Paperback Software International*, 87-0076-K (D.Mass.), at the Phase I Trial, in lieu of oral, direct examination, as if it were given live in Court herein.

4. The parties expressly waive any right to adverse direct examination, cross-examination or curative direct examination in Court of any witness, including those whose testimony is introduced by either party pursuant to the preceding paragraph. The parties have made their final designation of testimony and other proof, unless otherwise ordered by the Court.

5. The parties agree to withdraw any objection to the introduction of evidence on the ground of relevance that is based upon an asserted violation of the Order Regulating Jury Trial, entered September 30, 1992, but reserve the right to argue that any such evidence should be given no weight by the Court as trier of fact in reaching its determinations herein.

6. Each party agrees not to assert any objection to the other party's counter-designations of deposition testimony pursuant to Rule 32(a)(4), Fed. R. Civ. P., on the ground that the counter-designation is unnecessary to make the designated excerpt complete and not misleading.

7. The parties agree to a schedule for resolving any issue raised by Lotus' Supplemental Complaint as set forth in the agreement annexed hereto as Exhibit B. The parties further agree to attempt in good faith to agree upon a procedure for the resolution of disputed facts, if any, pertinent to determination of such issues, including, but not limited to, a procedure comparable to that set forth herein.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development Corporation

SO ORDERED:

ROBERT E. KEETON
Robert E. Keeton, U.S.D.J.

Date: 2/2/93

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

FILED IN OPEN COURT
Date: 2/2/93
Lowrie Deputy Clerk

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

IT IS HEREBY STIPULATED AND AGREED, by and between the undersigned counsel for plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland"), that:

1. The order of display of the long prompts within the "123-compatible" modes of Quattro and Quattro Pro follows the order of display of the menu commands within those modes, and each such long prompt provides a short textual description of the command to which it relates.

2. Lotus shall not contend, in this action or any appeal therefrom, that Borland has copied the long prompts of Lotus 1-2-3 in ~~any version of~~ Quattro or Quattro Pro [.] ~~published to date.~~

3. Borland shall not contend, in this action or any appeal therefrom, that Borland has not copied the long prompts of Lotus 1-2-3 in ~~any version of~~ Quattro or Quattro Pro [.] ~~published to date.~~

4. Neither party shall contend, in this action or any appeal therefrom, that the issue of whether or not Borland copied the long prompts of Lotus 1-2-3 in either Quattro or Quattro Pro is material to any other issue that has been or will be resolved in this case.

5. Nothing herein shall prevent either party from making reference to the stipulations set forth above, or to any findings of fact and conclusions of law that have been or will be reached by the Court concerning whether the menu commands, menu command structure, keystroke sequences and/or macro language of Lotus 1-2-3 have been copied in the "123-compatible" modes of Quattro and Quattro Pro.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development
Corporation

SO ORDERED:

ROBERT E. KEETON
Robert E. Keeton, U.S.D.J.

Date: 2/2/93

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

IT IS HEREBY STIPULATED AND AGREED, by and between the undersigned counsel for plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") that:

1. The deposition of Robert Kohn on subjects relating to his Supplemental Declaration, dated January 25, 1993, shall take place on February 10, 1993. Lotus shall notify Borland by February 5, 1993 if it intends to take this deposition. Lotus shall have the right to request one additional deposition during this time period, and Borland reserves the right to object to this request for deposition.

2. Lotus' memorandum regarding issues relating to the Key Reader shall be served on February 19, 1993. This memorandum may be accompanied by up to three Declarations (the "New Lotus Declarants") (not counting declarations of counsel presenting exhibits).

3. Borland shall have the opportunity to take the deposition of any New Lotus Declarant on February 25 and 26, 1993 (subject to the witnesses' availability), or as soon thereafter as practical.

4. Borland's memorandum in response to Lotus' memorandum shall be served on March 8, 1993, and may be accompanied by a single expert declaration responding to any expert declaration submitted by Lotus. Other than such an expert declaration (if Lotus first files an expert declaration) and any designations from the depositions of any New Lotus Declarants, Borland's evidentiary submissions are complete. Lotus may depose Borland's declarant on March 15, 1993 (subject to the witnesses' availability), or as soon thereafter as practical.

5. Lotus shall designate any portions of the transcript of the deposition referred to in paragraph 4 hereof it wishes to offer in evidence within two days of receipt of that transcript.

6. The parties shall jointly request the Court to schedule promptly thereafter a nonjury trial to resolve disputed issues of fact, if any, or an oral argument if the parties agree, or the Court determines, that no such issues exist.

7. Executing this stipulation does not constitute a waiver by either party of any right to present live testimony at any trial of the issues raised by the supplemental complaint.

8. The parties agree in good faith to negotiate and agree upon a factual stipulation concerning the Key Reader portion of the "QPW.EXE" file in the Quattro Pro for Windows product in lieu of disclosure thereof by Borland.

9. Both parties reserve the right to request the Court to modify the terms of this Stipulation for good cause shown.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development Corporation

SO ORDERED:

ROBERT E. KEETON
Robert E. Keeton, U.S.D.J.

Date: 2/3/93

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

IT IS HEREBY STIPULATED AND AGREED, by and between the undersigned counsel for plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland"), that:

1. The order of display of the long prompts within the "123-compatible" modes of Quattro and Quattro Pro follows the order of display of the menu commands within those modes, and each such long prompt provides a short textual description of the command to which it relates.

2. Lotus shall not contend, in this action or any appeal therefrom, that Borland has copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

3. Borland shall not contend, in this action or any appeal therefrom, that Borland has not copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

4. Neither party shall contend, in this action or any appeal therefrom, that the issue of whether or not Borland copied the long prompts of Lotus 1-2-3 in either Quattro or Quattro Pro

is material to any other issue that has been or will be resolved in this case.

5. Nothing herein shall prevent either party from making reference to the stipulations set forth above, or to any findings of fact and conclusions of law that have been or will be reached by the Court concerning whether the menu commands, menu command structure, keystroke sequences and/or macro language of Lotus 1-2-3 have been copied in the "123-compatible" modes of Quattro and Quattro Pro.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development
Corporation

SO ORDERED:

ROBERT E. KEETON
Robert E. Keeton, U.S.D.J.

Date: 2/3/93

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

STIPULATION AND ORDER REGULATING TRIAL

Plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") hereby agree as follows:

1. *Scope.* This Stipulation and Order is to govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial"). The parties' waivers of their respective rights to trial by jury are as set forth on page 63 of the January 14, 1993 hearing transcript.

2. The parties have entered into a stipulation concerning the "long prompts" in the parties' respective products (a copy of which is annexed hereto as Exhibit A), the terms of which are incorporated herein by reference.

3. The parties hereby consent to the introduction of designated affidavits, declarations or deposition testimony from this action or *Lotus Development Corp. v. Paperback Software International*, 87-0076-K (D.Mass.), at the Phase I Trial, in lieu of oral, direct examination, as if it were given live in Court herein.

4. The parties expressly waive any right to adverse direct examination, cross-examination or curative direct examination in Court of any witness, including those whose testimony is introduced by either party pursuant to the preceding paragraph. The parties have made their final designation of testimony and other proof, unless otherwise ordered by the Court.

5. The parties agree to withdraw any objection to the introduction of evidence on the ground of relevance that is based upon an asserted violation of the Order Regulating Jury Trial, entered September 30, 1992, but reserve the right to argue that any such evidence should be given no weight by the Court as trier of fact in reaching its determinations herein.

6. Each party agrees not to assert any objection to the other party's counter-designations of deposition testimony pursuant to Rule 32(a)(4), Fed. R. Civ. P., on the ground that the counter-designation is unnecessary to make the designated excerpt complete and not misleading.

7. The parties agree to a schedule for resolving any issue raised by Lotus' Supplemental Complaint as set forth in the agreement annexed hereto as Exhibit B. The parties further agree to attempt in good faith to agree upon a procedure for the resolution of disputed facts, if any, pertinent to determination of such issues, including, but not limited to, a procedure comparable to that set forth herein.

Dated: February 2, 1993

WILSON, SONSINI, GOODRICH & ROSATI

By: GARY L. REBACK
Gary L. Reback

Two Palo Alto Square
Palo Alto, CA 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, MA 02110

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: HENRY B. GUTMAN
Henry B. Gutman

30 Rockefeller Plaza
New York, NY 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, MA 02109

Attorneys for Plaintiff
Lotus Development Corporation

SO ORDERED:

_____ Date: _____
Robert E. Keeton, U.S.D.J.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

BORLAND'S MOTION TO AMEND ORIGINAL ANSWER

Defendant Borland International, Inc. ("Borland") hereby moves, pursuant to Fed. R. Civ. P. 15(a), for leave to file an amended answer (attached hereto as Exhibit A) that includes the affirmative defense of fair use. In the alternative, Borland moves, pursuant to Fed. R. Civ. P. 15(b), for permission to file an amended answer to conform to the evidence submitted at trial.

As set forth in the accompanying memorandum, Borland makes this motion on the grounds that its affirmative defense of fair use is based upon case authority that was promulgated very recently, and on the grounds that plaintiff Lotus Development Corporation ("Lotus") is not prejudiced by the assertion of the defense at this time.

Borland requests that this Court grant Borland leave to file an amended complaint and grant such further relief as the Court may deem appropriate.

REQUEST FOR ORAL ARGUMENT

Borland respectfully requests that a hearing on this motion pursuant to Local Rule 7.1(c) be held at the start of the upcoming phase of trial.

Dated: February 25, 1993

WILSON, SONSINI GOODRICH
& ROSATI

By PETER N. DETKIN
Peter N. Detkin

Two Palo Alto Square
Palo Alto, California 94306
(415) 493-9300

Peter E. Gelhaar
DONELLY, CONROY, & GELHAAR
176 Federal Street
Boston, MA 02110
(617) 261-9890

Attorneys for Defendant
BORLAND INTERNATIONAL, INC.

CERTIFICATE OF SERVICE

I hereby certify that the foregoing Motion and the Memorandum of Points and Authorities in Support of this Motion were served by overnight courier on counsel of record for Lotus Development Corp. on February 27, 1993.

[ILLEGIBLE]

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No.
90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

AMENDED ANSWER AND DEMAND FOR JURY TRIAL

Defendant Borland International, Inc. ("Borland") answers the complaint of Lotus Development Corporation ("Lotus") as follows:

FIRST DEFENSE

1. Borland admits that plaintiff purports to bring an action for copyright infringement arising out of Borland's sale of computer software programs called "Quattro" and "Quattro Pro." Except as expressly admitted herein, Borland denies each and every allegation contained in paragraph 1.

2. Borland admits that the Court has subject matter jurisdiction as conferred by 28 U.S.C. §§1331 and 1338. Borland denies each and every other allegation contained in paragraph 2.

3. Borland admits that it is doing business in this judicial district. Borland denies that venue is proper in this Court. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 3.

4. Borland admits that Lotus is a corporation organized and existing under the laws of the State of Delaware, with its principal place of business in Cambridge, Massachusetts. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 4.

5. Borland admits that its principal place of business is at 1800 Green Hills Road, Scotts Valley, Santa Cruz County, California. Borland further admits that it is engaged in the business of marketing and distributing personal computer programming language and application software programs on a nationwide basis, including within the District of Massachusetts. Borland denies the remaining allegations contained in paragraph 5.

6. Borland admits that there is a computer software program entitled "Lotus 1-2-3;" that Lotus 1-2-3 provides users of certain computers with the ability to perform certain computer-assisted tasks including the accumulation, arrangement, and calculation of data and text in what is known as a spreadsheet format; that Lotus 1-2-3 also contains certain database storage and management and graphics presentation capabilities; and that Lotus 1-2-3 is sold as part of a package which includes user instruction manuals and other documentation. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 6.

7. Borland admits that users of Lotus 1-2-3 communicate with Lotus 1-2-3 by various means. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 7.

8. Borland denies that Lotus 1-2-3 is copyrightable subject matter under the laws of the United States, at least to the extent apparently claimed by Lotus for the purposes of the present action. Borland is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations in paragraph 8.

9-15. Borland is without knowledge or information sufficient to form a belief as to the truth of the allegations within paragraphs 9, 10, 11, 12, 13, 14, and 15.

16. Borland denies each and every allegation contained in paragraph 16.

17. Borland admits that Quattro offers users the capability to create an unlimited number of alternative commands and offers several examples of command alternatives, and that Exhibit A to the Complaint is a copy of a chart that at certain times has been included in the Quattro "Reference Guide" provided with the Quattro program. Borland denies each and every other allegation contained in paragraph 17.

18-26. Borland denies each and every allegation contained in paragraphs 18, 19, 20, 21, 22, 23, 24, 25, and 26.

27. Borland admits that Lotus has made various statements regarding its copyright positions at various times. Borland denies each and every other allegation contained in paragraph 27.

28-29. Borland denies each and every allegation contained in paragraphs 28 and 29.

SECOND DEFENSE

Lotus has failed to state a claim upon which relief can be granted.

THIRD DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu structure and commands as alleged in the Complaint, at least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a valid copyright therein, because the Lotus 1-2-3 menu structure and commands constitute uncopyrightable subject matter.

FOURTH DEFENSE

Lotus does not have a valid copyright in the Lotus 1-2-3 menu structure and commands as alleged in the Complaint, at least to the extent Lotus alleges Borland has copied, published, distributed, marketed, or placed in interstate commerce any such features; nor can Lotus obtain a copyright therein, because such features lack originality.

FIFTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of laches. Borland has manufactured, used, and sold its Quattro products for a substantial period of time. Despite its knowledge of such activities, Lotus failed to assert the claims raised in the Complaint within a reasonable period of time. Borland relied upon this failure by Lotus and would be prejudiced were Lotus permitted now belatedly to assert the claims contained in its Complaint.

SIXTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of estoppel.

SEVENTH DEFENSE

Lotus' claims are barred in whole or in part by the doctrine of waiver.

EIGHTH DEFENSE

Borland has not copied, published, distributed, marketed, or placed in interstate commerce any computer software program which infringes upon or violates any rights of Lotus, including any rights that might exist under copyright.

NINTH DEFENSE

To the extent, if any, Lotus has a valid copyright in the Lotus 1-2-3 menu command hierarchy, macro language, and keystroke sequences, Borland's use thereof constitutes a fair use as allowed for in 17 U.S.C. § 107.

WHEREFORE, defendant Borland International, Inc. prays for judgment as follows:

A. That the relief requested by Lotus be denied and that the Complaint be dismissed with costs and attorneys' fees to defendant;

B. That such other and further relief be granted as the Court deems just and proper.

JURY DEMAND

Pursuant to Fed. R. Civ. P. 38(b), defendant hereby demands a trial by jury of all issues so triable.

Dated: February __, 1993

WILSON, SONSINI, GOODRICH
& ROSATI

By: _____
Gary L. Reback
Two Palo Alto Square
Palo Alto, California 94306
(415) 493-9300

Peter E. Gelhaar
DONELLY, CONROY, & GELHAAR
176 Federal Street
Boston, MA 02110
(617) 261-9890

Attorneys for Defendant
BORLAND INTERNATIONAL, INC.

CERTIFICATE OF SERVICE

I hereby certify that the foregoing Amended Answer and Demand For Jury Trial were served by overnight courier on counsel of record for Lotus Development Corp. on February __, 1993.

_____
Attorneys for Defendant
BORLAND INTERNATIONAL, INC.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,
Defendant.

NOTICE OF FILING OF TRIAL RECORD

Plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") have filed, concurrently herewith, the following evidentiary and other materials pertinent to the Phase I and Phase II Trials in this matter, held on February 1-3 and March 31-April 2, 1993, respectively:

I: *TRIAL BINDERS*

Two boxes containing 9 binders of material proffered and referred to at the Phase I and Phase II Trials, marked to indicate the party designating the material, together with objections thereto and rulings thereon:
Volume I, II and III—Deposition Testimony
Volume IV—Exhibits
Volume V—Lotus' Declarations
Volume VI—Borland's Declarations
Volume VII—Discovery Responses

Volume VIII—Pleadings and Transcripts
Volume IX (red)—Confidential Portion of Phase I and Phase II Trial Record

II: *OVERSIZED EXHIBITS*

Two boxes containing certain exhibits not included in the Trial Binders due to their size or nature, as indicated therein:

Lotus 1-2-3, Release 2.0 (software program) (Plaintiff's Exhibit 2)

PC Magazine, May 26, 1987 (Plaintiff's Exhibit 28)

Dybvig, Philip H., *The Lotus Tutorial* (The Scientific Press 1989) (Plaintiff's Exhibit 36)

McComb, Gordon, *Lotus 1-2-3 and the IBM PC or XT* (New American Library 1989) (Plaintiff's Exhibit 38)

Demonstration Disk containing files used by Larry Roshfeld in Lotus demonstration on March 31, 1993 (Plaintiff's Exhibit 49)

Product Packaging for Quattro Pro 2.0, Quattro Pro 3.0, Quattro Pro 4.0, Quattro Pro 4.01 and Quattro Pro for Windows (Plaintiff's Exhibit 50)

Portion of videotape submitted by Borland in connection with cross-motion for summary judgment on September 28, 1991 (Defendant's Exhibit 523)

Aho, Alfred, et al., *Compilers, Principles, Techniques and Tools* (Addison-Wesley 1988) (Defendant's Proposed Exhibit 506)

III: *SOFTWARE DOCUMENTATION*

Boxes containing documentation (*i.e.*, user's manuals) for the following computer software products included in the "Software Library" to be filed concurrently herewith:

Lotus 1-2-3, Release 1A
(Plaintiff's Exhibit 1)

Lotus 1-2-3, Release 2.01
(Plaintiff's Exhibit 3)

Symphony, Release 2.0
(Plaintiff's Exhibit 24)

Improv (Plaintiff's Exhibit 27)

Quattro (Plaintiff's Exhibit 4)

Quattro Pro 1.0 (Plaintiff's Exhibit 5)

Quattro Pro 2.0 (Plaintiff's Exhibit 6)

Quattro Pro 3.0 (Plaintiff's Exhibit 7)

Quattro Pro 4.0 (Plaintiff's Exhibit 8)

Quattro Pro 4.01 (part of Plaintiff's Exhibit 35; also Defendant's Exhibit 522)

Quattro Pro for Windows (part of Plaintiff's Exhibit 35; also Defendant's Exhibit 518)

Quattro Pro SE (Defendant's Exhibit 516)

Microsoft Excel, Version 2.1 (Reference Guide only) (Plaintiff's Exhibit 20)

Microsoft Excel, Version 3.0 (Defendant's Exhibit 519)

Microsoft Excel, Version 4.0 (Defendant's Exhibit 520)

Microsoft MultiPlan, Version 4.0 (Plaintiff's Exhibit 21)

Framework XE for DOS (Executive Edition)
(Plaintiff's Exhibit 22)

SuperCalc 3 (Plaintiff's Exhibit 23)*

SuperCalc 5 (Defendant's Exhibit 521)

MathPlan 2.1 (Plaintiff's Exhibit 25)

PFS Professional Plan Version 1.0
(Plaintiff's Exhibit 26)

* Plaintiff's Exhibit 23 consists of the SuperCalc 4 software program, which is included in the "Software Library" to be filed concurrently herewith. Lotus is endeavoring to locate the SuperCalc 4 documentation and will substitute it for this item, if possible.

Dated: May 4, 1993

Respectfully submitted,

WILSON, SONSINI,
GOODRICH & ROSATI

By: PETER N. DETKIN
Peter N. Detkin
(admitted *pro hac vice*)

Two Palo Alto Square
Palo Alto, California 94306
(415) 493-9300

DONNELLY, CONROY
& GELHAAR
176 Federal Street
Boston, Massachusetts 02110
(617) 261-9890

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV &
KARABELL

By: KERRY L. KONRAD
Kerry L. Konrad
(admitted *pro hac vice*)

30 Rockefeller Plaza
New York, New York 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, Massachusetts 02109
(617) 526-6000

Attorneys for Plaintiff
Lotus Development
Corporation

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,
Defendant.

NOTICE OF FILING OF "SOFTWARE LIBRARY"

Plaintiff Lotus Development Corporation ("Lotus") and defendant Borland International, Inc. ("Borland") have filed, concurrently herewith, the following computer software programs proffered in evidence and referred to in the Phase I and Phase II Trials in this matter, held on February 1-3 and March 31-April 2, respectively:

Lotus 1-2-3, Release 1A (Plaintiff's Exhibit 1)

Lotus 1-2-3, Release 2.01 (Plaintiff's Exhibit 3)

Symphony, Release 2.0 (Plaintiff's Exhibit 24)

Improv (Plaintiff's Exhibit 27)

Quattro (Plaintiff's Exhibit 4)

Quattro Pro 1.0 (Plaintiff's Exhibit 5)

Quattro Pro 2.0 (Plaintiff's Exhibit 6)

Quattro Pro 3.0 (Plaintiff's Exhibit 7)

Quattro Pro 4.0 (Plaintiff's Exhibit 8)

Quattro Pro 4.01 (part of Plaintiff's Exhibit 35; also Defendant's Exhibit 522)

Quattro Pro for Windows (part of Plaintiff's Exhibit 35; also Defendant's Exhibit 518)

Quattro Pro SE (Defendant's Exhibit 516)

Microsoft Excel, Version 2.1 (Plaintiff's Exhibit 20)

Microsoft Excel, Version 3.0 (Defendant's Exhibit 519)

Microsoft Excel, Version 4.0 (Defendant's Exhibit 520)

Microsoft MultiPlan, Version 4.0 (Plaintiff's Exhibit 21)

Framework XE for DOS (Executive Edition) (Plaintiff's Exhibit 22)

SuperCalc 4 (Plaintiff's Exhibit 23)

SuperCalc 5 (Defendant's Exhibit 521)

MathPlan 2.1 (Plaintiff's Exhibit 25)

PFS Professional Plan Version 1.0 (Plaintiff's Exhibit 26)

For the Court's convenience, the programs contained in the "Software Library" have been previously installed by the parties on a computer running the Microsoft Windows operating environment software. Selection of the appropriate icon from the pre-installed menu will provide access to the indicated program.

For technical reasons, the Lotus 1-2-3, Release 2.0 software program (Plaintiff's Exhibit 2) could not be installed on this computer's hard drive. The disks for that program are contained in the product package, which is included in the box of

"Oversized Exhibits" to be filed concurrently herewith. If the Court wishes, the parties will be happy to assist with the installation necessary to review that program.

Dated: May 4, 1993

Respectfully submitted,

WILSON, SONSINI, GOODRICH & ROSATI

By: PETER N. DETKIN
Peter N. Detkin
(admitted *pro hac vice*)

Two Palo Alto Square
Palo Alto, California 94306
(415) 493-9300

DONNELLY, CONROY & GELHAAR
176 Federal Street
Boston, Massachusetts 02110
(617) 261-9890

Attorneys for Defendant
Borland International, Inc.

O'SULLIVAN GRAEV & KARABELL

By: KERRY L. KONRAD
Kerry L. Konrad
(admitted *pro hac vice*)

30 Rockefeller Plaza
New York, New York 10112
(212) 408-2400

HALE AND DORR
60 State Street
Boston, Massachusetts 02109
(617) 526-6000

Attorneys for Plaintiff
Lotus Development
Corporation

[Volume 12, Page 1]

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION
No. 87-76-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

PAPERBACK SOFTWARE INTERNATIONAL and STEPHENSON SOFTWARE, LTD.,

Defendants.

CIVIL NONJURY TRIAL

TWELFTH DAY—FINAL ARGUMENTS

BEFORE THE HONORABLE ROBERT E. KEETON
United States District Judge

Courtroom 11
U.S.P.O. & Courthouse
Boston, Massachusetts
Thursday, March 22, 1990

Nancy L. Eaton, Reporter Computer-Aided Transcription
13 Short Street, Reading, MA 01867 (617) 944-7182

[Volume 12, Page 71]

is free for everybody to use. You just can't copy the compiler. That's protectable and that's certainly protectable by copyright. You have to have your own.

What they did here is they copied our compiler. Excel uses our macros. Your Honor saw it. But they did it by coming up with their own—well, they didn't do a compiler. They did a converter or translator instead. It is a different technique. They did it without copying. They have their own macro language. They didn't copy ours and they can read or use our macros by means of their converter, just as someone could with two different BASIC compilers or two different C compilers; so the analogy to a programming language doesn't help them and doesn't hurt us. All we're seeking there is to protect what anybody else could protect, even in a programming language, and what their own witness admitted is protected.

I would also just to clear up one point in this area that may have been confused during the trial. We don't claim that the macros people create belong to us. That's theirs. That's their property. Somebody writes a macro using Lotus 1-2-3, there is no question that that macro belongs to the user.

Now, there are a lot of references, your Honor, to buttons, levers, watches. Last night I reread Mr. Gupta's opening and I recalled the poignant moment when he

* * *

[Volume 12, Page 73]

there is nothing original there. His was one of those Japanese digital type that had the numbers there with the colon between them. There is no originality in that user interface.

If the face of that watch had had Spirew Agnew's picture on it, Spirew Agnew's depiction and the hands were his hands or Mick Mouse's or Minni Mouse's hands or pick your favorite character, that's copyrightable. There are cases on those. That's easy. Even a watch user interface can be protected.

I mean there is also the point, and I heard an echo of it again this morning, that somehow there is just not—I mean that the reason we're doing this business about the classifications and we're trying to argue so hard that this is a single work is because we don't want the menus to have to be judged standing alone, that there isn't enough subject matter, there isn't enough writing there, there isn't enough words for that to be copyrightable subject matter.

And I would suggest to the Court, as we did in our brief, if you don't look at it, A, that would be wrong even if you looked at it screen by screen; B, that's not the way you look at it. You look at the work as a whole. But I brought in today a poem which my six-year old wrote for school, and if I thought about it far enough in

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION

—vs.—

BORLAND CORPORATION

CIVIL ACTION No.
90-11663-K

Courtroom 11
Boston, Massachusetts
Tuesday, June 18, 1991

LOTUS DEVELOPMENT CORPORATION

—vs.—

THE SANTA CRUZ OPERATION, INC.

CONFERENCE

BEFORE THE HONORABLE ROBERT E. KEETON
UNITED STATES DISTRICT JUDGE

TIMOTHY J. WILLETTE, RPR-CM
Official Court Reporter
U.S. District Court
1201 USPO & Courthouse
Boston, Massachusetts 02109
(617) 248-0604

[4] THE COURT: Well, I think it makes sense to deal first with the motion for protective order and the dispute over discovery. Now, I have read your submissions on that matter and I take note of the fact that the Borland position is premised on a legal argument that I am prepared to rule is without merit.

Basically, as I understand your position, it is that copyrightability depends on patentability, and I've heard and considered that argument. I don't consider myself bound by what I ruled before if you could show me some [5] reason I made a mistake or that there's some intervening precedent, but having examined your submission, the memorandum in support of that motion to compel, most of which is on this argument, I take it that means you are premising the argument on this proposition that if you can show me that there's something patentable here that would mean there's no copyright protection. I think that is plainly error, and since that's the premise of the motion to compel and the effort to get additional discovery, I think I should simply deny the motion to compel and give you my ruling that I don't regard that matter as tending to produce the discovery of any evidence that is either admissible in this case or is likely to lead to admissible evidence.

Now, there is the additional proposition that I have also examined, the in camera submission, which of course you have not seen, but if I got over that first hurdle, I think I would still find it necessary to deny your motion to compel for reasons I'm not able to explain to you in detail at this point. But I don't really need to reach that issue, because on the first proposition — let me put it to you another way:

If you spend 90 percent of your brief on the matter arguing a legal proposition as the premise for the motion that is not sound, I take you at your representation that that's the premise of your motion and that you don't [6] have any other theory that's legally supportable for the discovery you're seeking, and so I think that's the ruling.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

Memorandum and Order
March 20, 1992

In this civil action, the plaintiff, Lotus Development Corporation ("Lotus"), seeks damages and equitable relief for alleged infringement by defendant, Borland International, Inc. ("Borland"), of the Lotus copyright in its computer software program, Lotus 1-2-3. This is the same copyright for the infringement of which Lotus has obtained relief under this court's decision in *Lotus Dev. Corp.* v. *Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990) (holding that Paperback's VP Planner was an infringing software product).

Lotus contends that on undisputed facts Borland's Quattro and Quattro Pro (herein, collectively, the "Quattro programs") are likewise infringing. Borland responds that its products are materially different from both Lotus 1-2-3 and VP Planner and that the court should determine on undisputed facts that the Quattro programs do not infringe the Lotus copyright. Borland argues that this result follows under the rules and standards of law applied

* * *

2(a)

* This footnote, not for publication, identifies for the parties and for the clerk's recordkeeping the submissions now before the court: (1) Lotus' Motion for Summary Judgment on Liability (Docket No. 30, filed May 10, 1991); (2) Borland's Cross-Motion for Summary Judgment on Copyrightability (Docket No. 87, filed September 30, 1991); (3) Brief Amicus Curiae on Behalf of the Register of Copyrights (Docket No. 85, filed September 30, 1991); (4) Borland's Statement of Material Facts Going to Copyrightability As to Which There Is No Genuine Issue (Docket No. 93, filed September 30, 1991); (5) Liability Questions for the Trier of Fact (Docket No. 95, filed September 30, 1991); (6) Brief Amicus Curiae of Copyright Law Professors (Docket No. 102, filed October 9, 1991); (7) Plaintiff's Memorandum of Law in Support of Motion for Summary Judgment on Liability (Redacted as of January 15, 1992) (Docket No. 147, filed January 28, 1992); (8) Statement of Material Facts As to Which There Is No Genuine Issue Pursuant to Local Rule 56.1 (Docket No. 149, filed January 28, 1992); (9) Plaintiff's Reply Memorandum in Support of Motion for Summary Judgment and in Opposition to Defendant's Cross-Motion for Summary Judgment on the Issue of Copyrightability (Docket No. 153, filed January 28, 1992); (10) Lotus' Response to Borland's Statement of Facts Going to Copyrightability As to Which There Is No Genuine Issue (Docket No. 154, filed January 28, 1992); (11) Borland's Memorandum in Support of Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment (Redacted as of January 15, 1992) (Docket No. 141, filed January 30, 1992); (12) Borland's Statement of Material Facts As to Which There Is No [sic] Genuine Issue to Be Tried (Docket No. 144, filed January 30, 1992); (13) Borland's Reply Memorandum in Support of Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment (Docket No. 146, filed January 30, 1992); (14) Brief of Amicus Curiae (of Software Entrepreneur's Forum) attached to Docket No. 59, filed June 17, 1991); (15) Plaintiff's Memorandum Regarding Issues for Jury Determination and Proposal for Phasing of Trial (Docket No. 50, filed June 7, 1991); (16) Borland's Redacted Memorandum of Law with Respect to Phasing of the Case (Docket No. 51, filed June 10, 1991); (17) Lotus' Response to Defendant Borland's Memorandum of Law with Respect to Phasing of the Case (no docket no., filed June 14, 1991); and (18) Borland's Reply Memorandum with Respect to Phasing of the Case (Docket No. 54, filed June 14, 1991). Other relevant submissions before the court include thirty-seven binders of exhibits, eight software packages, three videotapes, one magazine, numerous declarations, affidavits, and other exhibits, an evidentiary objection and response, and three motions and supporting documents relating to the preservation of confidential information.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION
No. 90-11662-K

Courtroom 11
Tuesday, May 19, 1992
Boston, Massachusetts

LOTUS DEVELOPMENT CORPORATION

vs.

BORLAND INTERNATIONAL, INC.

HEARING ON CROSS-MOTIONS
FOR SUMMARY JUDGMENT

BEFORE THE HONORABLE ROBERT E. KEETON
UNITED STATES DISTRICT JUDGE

APPEARANCES:

For the Plaintiff:
O'SULLIVAN, GRAEV & KARABELL
By: Henry B. Gutman, Esquire
Kerry L. Konrad, Esquire
Joshua H. Epstein, Esquire
Paul M. O'Connor III, Esquire
30 Rockefeller Plaza
New York, New York 10112

For the Defendant:
WILSON, SONSINI, GOODRICH & ROSATI
By: Gary L. Reback, Esquire
Peter N. Detkin, Esquire
Two Palo Alto Square
Palo Alto, California 94306

DONNELLY, CONROY & GELHAAR
By: Peter E. Gelhaar, Esquire
176 Federal Street
Boston, Massachusetts 02110

Court Reporter:
Timothy J. Willette, RPR-CM
Official Court Reporter
U.S. District Court
P.O. Box 347
Boston, Massachusetts 02101-0347
(617) 248-0604

[3] to proceed with my argument. Let's begin with the issue that I will call copying. And just so there's no confusion, what I mean by copying is what the Court referred to as substantial similarity in the evidentiary sense, and I'll try and use that word consistently today so there's no confusion. Our view is that there are no issues of fact here concerning copying. Let's begin with the question of what are the similarities and the differences between the programs.

I don't believe there's a triable issue of fact there, your Honor, because each of those similarities and each of those differences are evident from looking at the programs. There's nothing hidden here, there's nothing that requires some kind of a probing inquiry. It's just a matter of examining the products. Now, to be sure, the parties may describe these items differently, their definitions of them may be fuzzy or vague or inconsistent, suggesting that there's an issue of fact, but if the Court examines the products, indeed I believe if one examines the record as it stands now on summary judgment, one could compile an objective list of everything that's the same and everything that's different and none of those items are genuinely in dispute. There is nothing to try there.

THE COURT: Let me ask your perspective on whether in addition to having that list for comparison, as [4] you have described it, I need to be thinking about the issue that Borland has stated repeatedly in their submissions, that your claim is not for infringement of Release 2.0 but only for an earlier version and that—maybe I'm overstating Borland's argument—if they just copied it from 2.0, that doesn't make your case under your pleadings. If they haven't put it that way, they put it so close to that that they're asking me to think of it that way, I believe. And I'll give Mr. Reback an opportunity to correct me if I'm wrong.

MR. GUTMAN: There's a short and simple answer to that, your Honor, and that is a stipulation that they signed.

THE COURT: I'm aware of that stipulation. Why does that eliminate the problem—I gather it's an argument that you might have had a claim, but you didn't make it in your complaint and you're stuck with your complaint.

MR. GUTMAN: Well, first of all, your Honor, under the law I don't believe that's so. What we have in 2.01, in Release 2.01, is something which from the perspective of the user interface is absolutely identical, absolutely identical, to 2.0. It is a derivative of 2.0, which is in turn a derivative of 1A and 1, and therefore, under the law, we don't have to sue under each of those if [5] what we're suing for them copying is something that was in the earlier version, and all of this was, and they have so stipulated.

THE COURT: Well, now, you see, that is where you are trying to explain to me some language here, copying, substantial similarity, what's the relation. What you have just said would seem obvious if we were just talking about similarities, but if we're talking about copying and similarity is simply one way of approaching evidence of copying, then how do you answer the argument that if your proof doesn't establish that they copied from the product on which your complaint is based there's a failure of your proof to match your claim?

MR. GUTMAN: Well, your Honor, 2.01 is identical to 2.0. It's like talking about the first printing and second printing of a book. Now, whether one views it as a question of copying or of substantial similarity, anything that can be said of a factual nature with respect to 2.01 can and must be said with respect to 2.0 as well. They are the same for these purposes, as Borland has stipulated.

THE COURT: What puzzles me is why you don't just ask me to allow you to amend your complaint and claim copying of 2.01 as well.

MR. GUTMAN: That's my fallback, your Honor.

[6] If we had to, obviously we would ask for leave to amend to conform to the proof.

THE COURT: Well, why don't you? Is there some other collateral consequence that you're worrying about?

MR. GUTMAN: Absolutely not. your Honor. It's simply Borland screaming about being surprised by this at the last minute, frankly their version of the story in which they say, "Oh, no, we were copying 2.01, not the identical 2.0"—

THE COURT: Is it correct that if your claim included copying from 2.01, that would not have affected the scope of discovery at all in this case, the scope of the issues?

MR. GUTMAN: Not at all, your Honor. We have in this case and in *Paperback* at every instance given full discovery with respect to 2.01, because for our purposes 2.01 and 2.0 were the same. We never drew a distinction, ever. And although they made such a claim in their brief, there is no support for that in the record, zero, and they have cited none. We produced the full discovery record concerning the development of 2.01 in the *Paperback* case, in the *Paperback* case, your Honor, and all of that discovery was provided to them here. They have never been blocked from asking questions about 2.01 here.

* * *

[44] My client has asked me to clarify this and I think it is appropriate that I do so.

Mr. Gutman repeatedly said that the idea was to be like Lotus. The idea of the native mode in both Quattro and Quattro Pro was to run user files that were created under 1-2-3. That was the idea. It was also Lotus' idea with respect to version 2.01, so I just want to make sure. That is the statement that Mr. Bosworth testified to, Mr. Warfield testified to, and Ms. Joanne Lin testified to. That was what they were trying to do. That was part of the idea, as Mr. Manzi said as well. Thank you, sir.

THE COURT: All right. Now, I will take the cross-motions under advisement. I think while you're here, though, we better schedule a time for a trial since regardless of how I rule here, with one possible exception, we'll have to have further proceedings. And I remain essentially of the view that the most likely form of those proceedings would be a form of trial on either a consolidated substantial similarity and copyrightability or successive, immediate. I'm not at all attracted to the idea that we put these questions to different juries. It seems to me the argument for that that you've advanced in

this brief is neither sound from the point of view of the legal issues involved, legal and factual issues involved, nor as a practical matter for the management of the case.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.
Defendant.

ORDER REGULATING JURY TRIAL
SEPTEMBER 30, 1992

I.

1. This case is set for jury trial, Phase I, commencing at 9:00 a.m. on November 2, 1992 (the TRIAL DATE).

OTHER IMPORTANT DATES:

[ONE] WEEK before the Final Pretrial Conference/October 16, 1992, 3:30 p.m.
See III-7(a)

Final Pretrial Conference October 16, 1992, 3:30 p.m.
See IV

[TWO] WEEKS before the TRIAL DATE/_____, 19__
See III-8(a), III-9

[ONE] WEEK before the TRIAL DATE/_____, 19__
See III-8(b)

TWO COURT DAYS before the TRIAL DATE/_____, 19_
See I-3, III-8(b), V-1, VI-A-2,
VI-B-1-(b), VII-A-2

TWO COURT DAYS BEFORE EACH DAY OF TRIAL
See VII-B-1

BEFORE TRIAL COMMENCES
See VII-D-1-(b)

DAILY DURING TRIAL
See VII-B-5, VII-E-1

2. Unless otherwise ordered after the date this ORDER is entered, Parts II-VIII of this ORDER will apply to the trial of this case.

3. The court does not press parties to settle a case that they genuinely prefer to try. If, however, this is a case destined for settlement, it is in the mutual interest of the parties, as well as in the public interest, that it be settled before the parties incur the expenses of final preparations for trial. Under this ORDER, intensified preparations commence, at the latest, about three weeks before the TRIAL DATE. If the case is not settled by that time, the parties and counsel are expected to make good faith efforts to determine finally whether the case can be settled *not later than TWO COURT DAYS before the TRIAL DATE.* These instructions do *not* mean that the court discourages continued efforts to settle a case *after the jury has been selected.* The court recognizes that developments in trial, beginning with party assessments of the composition of the jury, may affect demands and offers. The point the court emphasizes is that a settlement that occurs in the brief period of two court days before jury selection is to commence is clear evidence that the parties and their attorneys have deferred serious efforts to settle, with insensitivity to the waste of public and private resources and to the delay and inconvenience that their deferring serious negotiations has

caused to the parties, witnesses, and attorneys in other cases, as well as to the court.

II. Aims, Incentives, and Stipulations

1. The central aim of this Order is to create a set of procedures tailored to fit the distinctive characteristics of this case and "to secure the just, speedy, and inexpensive determination of [this] action," Fed. R. Civ. P. 1.

2. Absent planning and an explicit understanding among counsel and the court about methods of proof, interrogation, and argument that will or will not be used, each advocate has an incentive toward extremely adversary strategies and tactics. Of course, lawyers as well as judges know that extreme adversariness has its own downside risks. We also know, however, that pressures to respond in kind to contentious techniques used against you are hard to resist. Thus, when counsel expect that the court will allow excessive adversariness, the length and cost of the trial tend to increase. Distracting disputes over tangential matters interfere with the court's and the jury's understanding of material issues. The quality of the trial deteriorates.

3. When counsel and the court plan in advance to adapt trial procedures to the needs of the particular case, the trial is likely to be shorter and less expensive to the parties and to the public than it otherwise would be. Of even greater significance, the trial is likely to be better in quality. A crisp, well-focused trial helps the court and jury understand fully the material disputes of fact and law. The result is more likely to be a wise and fair decision on the merits.

4. The court encourages stipulations that will serve the aim that

> "the mode and order of interrogating witnesses and presenting evidence" be such as will "(1) make the interrogation and presentation effective for the ascertainment

> of truth, (2) avoid needless consumption of time, and (3) protect witnesses from harassment or undue embarrassment,"

Fed. R. Evid. 611(a). The kinds of agreements worthy of consideration include stipulations that direct testimony of some or all witnesses will be taken in narrative or affidavit form (with rights of cross-examination reserved) rather than "orally in open court" (as is the right of each party under Fed. R. Civ. P. 43(a), absent stipulation), and that evidence in affidavit form will be read to the jury by the witness, or by counsel or another reader with court approval. The court encourages counsel and the parties to consider also a stipulation for shorter time limits than those the court is likely to impose pursuant to Part III of this ORDER.

5. By entering this Order Regulating Trial, the court gives notice regarding rules of proof and procedure it expects to apply in the absence of stipulation and invites the parties to suggest modifications and additions to Parts V-VII of this ORDER that may more effectively tailor this trial to the needs of this case.

III. Time Limits[1]

1. Time limits provide an incentive to make the best possible use of the limited time allowed. If the parties are not able to agree upon time limits for the trial of this case, the court will, after inviting submissions from the parties, order

---

[1] Fed. R. Evid. 102 authorizes the trial judge to act to eliminate "unjustifiable expense and delay." Fed. R. Evid. 403 recognizes the power and duty of the court to exclude cumulative evidence or other evidence that takes more time than its probative value justifies. Fed. R. Evid. 611 directs the court to "exercise reasonable control over the mode . . . of interrogating witnesses and presenting evidence" to "avoid needless consumption of time." Fed. R. Civ. P. 1 expresses the aim of "just, speedy, and inexpensive determination of every action." The court invites the help of counsel in causing this trial to measure up to the spirit as well as the letter of these rules and the many precedents implementing and underscoring them.

presumptive limits, which will be subject to modification for good cause shown. Plainly, however, the limits that the court may order will not be as stringent as those the parties might agree would serve their mutual interests in achieving a shorter, less expensive, and better quality trial.

2. Absent agreement of the parties to time limits approved by the court, the court will order a presumptive limit of a specified number of hours for this trial, to be allocated equally between opposing parties unless otherwise ordered for good cause.

3. A request for added time will be allowed only for good cause. An explicit purpose of this provision is to create an incentive for using time exclusively on issues material to disposition on the merits.

4. In determining whether to allow a motion of any party for an increased allotment of time, the court will take into account (a) whether or not that party has used the time from commencement of trial forward in a reasonable and proper way, in compliance with all orders regulating the trial, (b) the party's proffer with respect to the way in which the added time requested would be used and why it is essential to fair trial, and (c) any other facts the party may wish to present in support of the motion, if determined by the court to be material. The court will be receptive to motions for reducing or increasing allotted time to assure that allotments are fair among the parties and adequate for developing the evidence. Any party that makes only proper use of its time throughout the trial is assured that an extension will be allowed if more time is needed to present all its material and admissible evidence adequately.

5. Presumptive allotments of time to a party will be stated as a total number of hours available to that party, rather than allocations of times for particular witnesses or proceedings. Thus, each party will be free, without a showing of good cause, to allocate time as that party chooses among different uses—opening statement, direct and cross-examination of various witnesses, closing argument, objections, and motions—as long as the party's total allotment is not exceeded.

6. Time taken to argue objections will be charged against the time allocation of the party against whom the court rules, and will be allocated between parties if the court rules partly for and partly against the objecting party.

7(a). Not later than *ONE WEEK before the Final Pretrial Conference*, each party shall file a draft of the precise text of each interrogatory it proposes that the court submit to the jury.

7(b). No evidence will be received in Phase I except evidence that is material to a disputed and material issue of fact for which counsel proffering the evidence (1) has filed with the court, at least one week before the Final Pretrial Conference, a proposed interrogatory to the jury, or (2) identifies an interrogatory proposed by the opposing party to which the proffered evidence is material.

8(a). Not less than *[TWO] WEEKS before the TRIAL DATE*, each party shall serve on the opposing party and file its NOTICE OF DIRECT EXAMINATION (1) listing its witnesses and an estimate of the time to be used in direct examination of each witness, (2) listing the precise pages and lines of any deposition testimony to be offered during the case in chief, with time estimates for reading that testimony into evidence, (3) affidavits of any expert witnesses whose depositions have not been taken, fairly summarizing the substance of their expected testimony, fully disclosing every opinion to be expressed, and estimating the time of direct examination, and (4) listing all the exhibits it intends to offer and an estimate of time, if any, to be used in publishing each exhibit to the jury. If the expected content of direct examination and exhibits has not previously been disclosed, the NOTICE shall include a fair summary of the content of each direct examination and each exhibit.

8(b). *Not less than [ONE] WEEK before the TRIAL DATE*, each party (or group of parties) shall serve and file its NOTICE OF CROSS-EXAMINATION estimating time to be used in cross-examination of each of the opposing party's listed witnesses. If either party, after seeing the opposing party's NOTICE OF DIRECT-EXAMINATION, proposes to call additional witnesses or offer additional exhibits, it shall, when serving and filing its NOTICE OF CROSS-EXAMINATION, also serve and file a SUPPLEMENTAL NOTICE of DIRECT EXAMINATION, including time estimates. An opposing party's SUPPLEMENTAL NOTICE OF CROSS-EXAMINATION shall be filed not later than TWO COURT DAYS BEFORE THE TRIAL DATE.

9. The parties are encouraged to confer and agree upon witness and exhibit lists and time limits for direct and cross-examination, and to file a stipulation *not less than TWO WEEKS before the TRIAL DATE* in lieu of the separate submissions otherwise required by paragraphs 7 and 8.

IV.

A final Pretrial Conference is scheduled for October 16, 1992 at 3:30 p.m.

V. Jury Selection

1. Not later than TWO COURT DAYS before the TRIAL DATE, the parties shall file a list of any other persons, not appearing on witness lists, who should be identified in voir dire questions to the jury. The list should be over-inclusive rather than under-inclusive in case of any doubt, in order to avoid risks of loss of jurors during trial because of acquaintance with a person whose possible relationship to the case was not made known during voir dire.

2. A jury of 8 will be selected unless the parties stipulate otherwise, with court approval.

3. It is the regular practice in this court to call a jury pool on Monday, or on Tuesday when Monday is a holiday. Jury selection in one or more other cases may be scheduled for the same TRIAL DATE. See Part VII-A-1, *infra*.

VI. Proposed Jury Instructions

A. Preliminary and Interim Instructions

1. Before testimony begins, the court will instruct the jury on the functions and roles of the jury and of counsel in the case and on the jury's obligations to decide the case solely on the evidence presented, to refrain from discussing the case (with each other or anyone else), and to avoid contact with the parties and with published or broadcast accounts of the trial.

2. The court may also give preliminary instructions *on the law applicable to the claims and defenses in this case*. The court will offer the parties an opportunity to be heard before giving preliminary instructions of this kind. Any requests of the parties for preliminary instructions shall be filed with the court *not later than TWO COURT DAYS before the TRIAL DATE*.

3. If the trial lasts longer than a week and the circumstances warrant, the court may give interim instructions from time to time to help the jury understand proceedings.

B. Final Jury Instructions

1. The court will give the final charge orally and ordinarily will also deliver a copy to the jury in writing. The charge will consist of four components:

(a) General Instructions: General Instructions serve as a guide to the jury throughout its deliberations. A draft of the court's proposed general instructions will be distributed in advance. Any objections or proposed amendments must have been filed on or before a date to be specified during trial. (_____, 19__.)

(b) Special Interrogatories: The court does not expect to ask the jury to return a general verdict. Special interrogatories request the jury's findings on specific questions of fact. Initial requests for questions to be included in the Verdict Form shall be served and filed as required in III-7(a). For good cause shown, revised requests may be served and filed *not later than TWO COURT DAYS before the TRIAL DATE.*

(c) Explanatory Instructions on the Law: Most of the explanatory instructions on the law bear directly upon an identified question or questions submitted in the special interrogatories on the Verdict Form. Ordinarily the court explains only those rules of law the jury needs to understand to answer the interrogatories. The court does not give, along with interrogatories, the type of instructions that are needed when the jury is to return a general verdict.

(d) Limiting Instructions: Limiting instructions may include instructions as to evidence received for a limited purpose or purposes, or against less than all the parties in the action. If the occasion for a limiting instruction can be anticipated, parties will be expected to have their requests prepared in advance in writing. If any evidence is received for a limited purpose, a party seeking the benefit of a limiting instruction in the court's final instructions will have the burden of assuring that a copy of the court's oral instruction is delivered to the Clerk for inclusion in the final charge, and in the case of documentary evidence, for attachment to the exhibit. A form that may be used with exhibits is attached to this order as "Exhibit A."

C. Jury Deliberation: Unless a stipulation to the contrary is filed, the Verdict must be unanimous. (The court encourages the parties to stipulate before jury selection that if excuses reduce the jury to a number not less than five, the remaining jurors will constitute the jury and will deliberate.)

VII. Procedure at Trial

A. Opening Statements

1. Opening statement by plaintiff will occur promptly after jury selection, on the TRIAL DATE, unless proceedings in another case have priority on that date.

2. Opening statement by the defendant will occur immediately after plaintiff's opening statement, unless defendant has elected otherwise by notice filed and served not later than TWO DAYS before the TRIAL DATE.

3. In a lengthy trial, the court may allow Interim Statements from time to time to enable counsel to clarify issues for the jury.

B. Evidence

1. Each party shall give advance notice to the court and the other parties, before jury selection, of the identity of all witnesses whose testimony (by affidavit, by deposition, or by oral testimony in trial) it may offer during trial. *Not later than TWO COURT DAYS before* it seeks to use the testimony of any witness, or on shorter notice for good cause shown, it shall advise the court and all other parties of its intent to use the testimony of the witness on the specified day. Except for good cause shown, no party shall be allowed to use the testimony of a witness other than the witnesses already listed on the filings with the court before trial commences. Except for good cause shown, no party shall introduce during direct examination documentary evidence other than those exhibits already listed with the court and furnished to the other parties before trial commences. These provisions with regard to documentary evidence shall not apply to cross-examination.

2. Absent a showing of good cause, the court will not exercise its discretion under Fed. R. Evid. 611(b) to allow the subject matter of the cross-examination to extend beyond the subject matter of the direct examination and matters affecting

the credibility of the witness. A showing of good cause will also be required if the subject matter of the redirect is to be allowed to extend beyond matters covered on cross-examination. That a witness has come from a distance or will be unavailable later in the trial may be found to constitute good cause to allow a party to treat him or her as its witness during what would otherwise be cross-examination, and to extend the examination beyond the scope of direct. *Absent a showing of special cause, examination of a witness shall not proceed beyond one redirect and one re-cross.*

3. Use of Depositions at Trial:

Except for good cause shown no deposition testimony shall be introduced as direct examination, or during oral direct examination, other than those pages or portions thereof noted in previous filings with the court. This limitation shall not apply to the use of deposition testimony in cross-examination.

4. Stipulations may be read at any time, unless otherwise ordered in a particular instance upon a showing of good cause.

5. At least one-half hour before commencement of trial each day, counsel shall furnish the court reporter with a copy of any document from which counsel intends to read that day, except depositions to be read by two people in question and answer form. Documents to be used during cross-examination are excepted.

6. Whenever a single person is reading deposition testimony, in order to enable jurors and the reporter to understand clearly, the reader will say "Question" before each question is read and "Answer" before each answer is read.

7. All documents or other non-testimonial evidence that will be admitted against at least one party without objection will be pre-marked as *numbered* exhibits. To effect the pre-marking and to avoid duplicative numbering, each of the parties will assign consecutive numbers to these documents and

objects, as follows: Plaintiff, 1-500; defendant 501-999. *The term "Exhibits" shall be used only* for documents and objects that are to be received without objection or have been received in evidence over objection.

8. The term "Marked Items" will be used for documents and other items, referred to in the proceedings, that are not Exhibits. A lettering system will be used by each of the parties to pre-mark as "Marked Items," for identification purposes, each piece of non-testimonial evidence it will offer to which objection has been made by the party against whom the item is sought to be admitted. The Clerk will supply the parties with stickers to be used in pre-marking documents and other non-testimonial evidence, either as agreed Exhibits or, for identification purposes, as Marked Items.

9. Counsel have the court's permission at all times to interrupt proceedings merely to object or move to strike. Counsel need not state the ground(s) of objection unless the court asks for the ground(s), but counsel may without invitation by the court state the ground(s) merely by reference to a Rule designated by number, among the Federal Rules of Evidence. Also, unless otherwise ordered (as may be done, for example, when the court interrupts to sustain an objection because there are obvious, valid grounds), counsel may state the grounds in customary legal jargon (*e.g.*, "hearsay," "irrelevant," "lack of essential foundation"). Counsel are not to go beyond a bare statement of the ground(s); supporting or opposing arguments will not be stated in the hearing of the jury without the court's permission.

10. Offers of proof will ordinarily be received only after the jury has been excused for a recess or for the day.

11. Conferences out of the hearing of the jury will be held to a minimum. *They will never occur at the beginning of a court day unless that timing is unavoidable.* When the court has directed jurors to be present at a designated hour, counsel asking for a conference out of the hearing of the jury at that

hour will be required to show good cause why the need should not have been anticipated so the jury could have been released early the preceding day and why the conference cannot be deferred until the end of the current day, or at least until the next recess.

12. Short conferences out of the hearing of the jury may be held at the side bar farthest from the jury box. The jury will be sent to the jury room if a more extended conference out of their hearing is required.

13. The objection of interrogating counsel to an answer that is nonresponsive will usually be sustained. Objections by other counsel solely on the ground that an answer is nonresponsive will usually be overruled. Sustaining such an objection is likely to lead to a new question that elicits exactly the same information as was stated in the excluded answer, and time is wasted. Of course, if some other valid ground of objection is added, a statement that the answer was nonresponsive may be needed and appropriate to explain why no objection was made to the question.

14. The court will not instruct a witness to "answer YES or NO" to (1) a multiple question, (2) a question that requires the witness to make or accept an inference or characterization rather than merely acknowledging or denying an observable fact, or (3) a question that is argumentative in form or in substance.

15. Questions framed to have more impact as arguments than as requests for testimony that the witness is competent to give are out of bounds. They will be excluded on objection and may be excluded on the court's initiative, without objection.

16. Ordinarily, questions asking one witness to comment on the credibility of another are out of bounds. A lawyer who wishes to ask such a question shall make a request out of the presence of the jury for leave to do so.

C. Schedule

1. The court will aim for conducting this trial 9:00 a.m. to 4:00 p.m. Monday-Friday.

2. There will be no trial of this case on the following days: [Holidays and other days specially committed].

D. Sequestration of Witnesses

If any party so requests, the following rules regarding sequestration will be enforced:

1. No person who is expected to testify as a witness in this civil action shall be present in the courtroom during the presentation of evidence except as follows:

(a) Professional persons engaged by a party or its counsel for the purpose of offering testimony as witnesses having specialized knowledge or experience may be present whenever evidence is being received, unless otherwise ordered.

(b) One representative of each party, designated by counsel to the court in advance of the trial as that party's representative, may be present throughout the trial.

2. A person who has testified and who is not expected to be called again by any party may be present in the courtroom after his or her testimony has been completed, but that person shall not state or summarize his or her own testimony or the testimony of others to prospective witnesses.

3. Counsel shall not state or summarize the testimony of others to prospective witnesses (other than professional persons within the group described in paragraph VII-D-1-(a) above) and shall not permit a prospective witness (other than a VII-D-1-(a) witness) to read transcripts of prior testimony of other witnesses.

E. Miscellaneous Matters

1. Documents filed in court during trial: A party filing a document in court rather than in the Clerk's Office must file, with the original, a copy of the first page. All documents will be given a docket number by the clerk.

2. Jurors may be permitted to take notes. If notetaking is allowed, instructions will be given in the form of Exhibit B.

VIII. DETERMINATIONS OF FACT AND CONCLUSIONS OF LAW

A. The following legal and factual issues (having been conclusively established in this case by Orders of March 20, 1992 and July 31, 1992 and accompanying Memoranda) will not be the subject of dispute at trial.

1. Borland copied into its Quattro programs the menu commands, menu hierarchy, macro language, and keystroke sequences of Lotus 1-2-3.

2. The menu commands, menu hierarchy, macro language, and keystroke sequences of Lotus 1-2-3 have copyrightable aspects.

a. The "idea" of the Lotus 1-2-3 user interface appropriate for analysis of the idea-expression dichotomy is that the user interface involves a system of menus, each menu consisting of less that a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.

b. The menu commands and menu hierarchy of Lotus 1-2-3 have expressive aspects and are copyrightable.

c. The Lotus 1-2-3 menu hierarchy was not limited to one or only a few possible arrangements at the time of Lotus' authorship.

d. The macro language and keystroke sequences of Lotus 1-2-3 are copyrightable to the extent that they incorporate copyrightable aspects of the menu commands and menu command hierarchy.

e. The expressive aspects of the menu commands, menu command hierarchy, macro language, and keystroke sequences are not trivial.

3. The emulation interfaces of the Quattro programs are substantially similar to Lotus 1-2-3, so that it is clear that Borland's copying of elements of 1-2-3 was illicit.

B. The following legal rulings limit the scope of evidence to be received at trial.

1. Evidence relevant to one or both of Borland's defenses of laches and estoppel that is not also relevant to a legal element of Lotus' prima facie case (or Borland's defense against Lotus' prima facie case) will not be received in the presence of the jury during Phase I of trial (in which the scope of Borland's infringement will be determined).

2. Substantial similarity in the mixed law-fact sense has been established and will not be the subject of further evidence at trial, even in relation to elements of the Quattro programs' emulation interfaces not yet determined to have been copied from Lotus or copyrighted by Lotus.

3. Evidence relating to elements of the Quattro programs *not* copied from Lotus will be excluded.

4. Lotus has not waived its right to proceed in this lawsuit.

C. The following questions (in terms of subject matter, though *not* form) will be the subject of Phase I of trial:

1. Did Borland copy the long prompts of Lotus 1-2-3?

2. Do the long prompts of Lotus 1-2-3 contain expressive elements that do not merge with the function of the long prompts—that is, do the long prompts have expressive elements not dictated by the need to explain the operations performed by associated menu commands and the relatively short length of the "long" prompts?

3. To what extent, if any, did functional constraints limit the number of possible ways in which the Lotus 1-2-3 menu command hierarchy could have been arranged at the time of its creation?

D. Evidence relating solely to Borland's defenses of laches and estoppel and to appropriate remedies for Borland's infringement of Lotus 1-2-3 shall not be received in Phase I of trial but may be received in a later phase or phases.

ROBERT E. KEETON
United States District Judge

EXHIBIT A

EXHIBIT MARKING SLIP

The attached document or object is Exhibit No.____.

Instructions to the Jury:

You may consider this document or object as evidence only with respect to any party whose name is checked below. You may not consider this document or object as evidence with respect to any party whose name is not checked. If any limited purpose is set forth below then you may only consider this document or object for that limited purpose. If no limited purpose is set forth below, then you may consider this document or object for all purposes as between the parties whose names are checked.

Party	*Limited Purpose*
____ Plaintiff(s) _____	_____
____ Defendant(s) _____	_____

EXHIBIT B

INSTRUCTIONS TO JURORS ON NOTE-TAKING

Ladies and Gentlemen of the Jury:

You have the permission of the court to take notes during the evidence, the summations of attorneys at the conclusion of the evidence, and during my instructions to you on the law.

In many courts—probably in most—jurors are not permitted to take notes. The reasons are concerned with fear that taking notes may cause the jury, as a whole, to be less effective in serving as a completely fair and impartial factfinder. Because of the potential usefulness of taking notes, you will be permitted to take notes in this trial. However, for the purpose of protecting against the possible disadvantages that have led many courts to order that notes not be taken, I will instruct you to observe the following limitations:

1. *Note-taking is permitted, not required.* Each of you may take notes. No one is required to take notes.

2. *Take notes sparingly.* Don't try to summarize all of the testimony. Notes are for the purpose of refreshing memory. They are particularly helpful when dealing with measurements, times, distances, identities, and relationships.

3. *Be brief.* Over-indulgence in note-taking may be distracting. You, the jurors, must pass on the credibility of witnesses; hence, you must observe the demeanor and appearance of each person on the witness stand to assist you in passing on his or her credibility. Note-taking must not distract you from that task. If you wish to make a note, you need not sacrifice the opportunity to make important observations. You may make your note after having made the observation itself. Keep in mind that when you ultimately make a decision in a case you will rely principally upon your eyes, your ears, and your mind, not upon your fingers.

4. *Your notes are for your own private use only. Do not use your notes, or any other juror's notes, as authority to persuade fellow jurors.* In your deliberations, give no more and

no less weight to the views of a fellow juror just because that juror did or did not take notes. Your notes are not official transcripts. They are personal memory aids, just like the notes of the judge and the notes of the lawyers. Notes are valuable as a stimulant to your memory. On the other hand, you might make an error in observing and you might make a mistake in recording what you have seen or heard. You are not, therefore, to use your notes as authority to persuade fellow jurors of what the evidence was during the trial.

5. *Do not take your notes away from court.* At the end of each day, please place your notes in the envelope which has been provided to you. A court officer will be directed to take the envelopes to a safe place and return them at the beginning of the next session on this case, unopened. At the conclusion of the case, after you have used your notes in deliberations, they will be collected and destroyed, to protect the secrecy of your deliberations.

Robert E. Keeton
United States District Judge

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

COURTROOM 11
WED., FEB. 3, 1993
Boston, Massachusetts

LOTUS DEVELOPMENT CORPORATION

—VS.—

BORLAND INTERNATIONAL, INC.

NON-JURY TRIAL PROCEEDINGS
DAY 3

BEFORE THE HONORABLE ROBERT E. KEETON
UNITED STATES DISTRICT JUDGE

APPEARANCES:

For the Plaintiff: O'SULLIVAN, GRAEV & KARABELL
By: Henry B. Gutman, Esquire
Kerry L. Konrad, Esquire
30 Rockefeller Plaza
New York, New York 10112

For the Defendant: WILSON, SONSINI, GOODRICH & ROSATI
By: Gary L. Reback, Esquire
Peter N. Detkin, Esquire
Two Palo Alto Square
Palo Alto, California 94306

DONNELLY, CONROY & GELHAAR
By: Peter E. Gelhaar, Esquire
176 Federal Street
Boston, Massachusetts 02110

Court Reporter: Timothy J. Willette, RPR-CM
Official Court Reporter
U.S. District Court
P.O. Box 347
Boston, Massachusetts 02101-0347
(617) 248-0604

[3-42] Lotus lay in the weeds, as it were, and sprung the trap shut at a point when Borland had no options. Had Lotus made its claim known early, Borland could have done exactly what the Lotus executives feared Borland would do. Borland could have changed its product to make it non-infringing. I hope on this point, your Honor, that your Honor will read the decisions that we cite and consider them in your Honor's deliberations of this matter. We have repeatedly cited the *Haas* case to Lotus and Lotus has yet to join us on the merits of that case. They make no comment on it whatsoever. Perhaps Mr. Gutman will comment on it today, but that is the case upon which we primarily rely.

Let me move—I see I'm taking up a lot of time, your Honor. Let me move now to the next point I want to make. In the January 14th hearing, your Honor, your Honor asked Mr. Gutman about the application of fair use to this case. That's between pages 47 and 49 of the transcript. Your Honor raised a number of questions with Mr. Gutman on fair use. Mr. Gutman answered them at that time and then submitted as part of the record without opposition by us several letters that were intended to cure the kinds of questions that your Honor had asked. And as we considered what your Honor referred to, we went back to California and we pulled the very recent cases that your Honor was referring to. Primarily it would appear that your Honor was [3-43] referring to *Sega vs. Accolade*, and that was just published and rendered final on January 3rd of this year. And we considered the fair use points that your

Honor was making and we thought that those points were meritorious and we thought we would have considerable common ground with the Court on fair use, although we had not had that common ground on copyrightability. So when we filed our answer to the supplemental complaint we pled fair use that your Honor had suggested going to all the issues in the case, and we brief it very briefly in the submission more as a request to your Honor to give us guidance as to what more your Honor wants.

There is a very small body of case law with respect to fair use. I think your Honor is familiar with it, but the primary cases that are applicable here are the *Sega vs. Accolade* case in the Ninth Circuit and the *Galoob* case both in the Ninth Circuit and the District Court. All those are very recent cases. And the point is here that Quattro and Quattro Pro's use of Lotus menu command hierarchy is a fair use either in Key Reader or on the screen. The point we would argue and submit to the Court is that the organization or order of commands is necessary to run macros and we have and we will proffer in the record Mr. Gutman's examination of our expert, Mr. Liddle, for that proposition, and that the words on the screen are necessary **[3-44]** for some interactive macros and for the writing, debugging, modifying and recording of macros.

Now, as I say, your Honor, I think we have common ground with the Court here, because we understand that this also is addressed to the Court's equity. We have previously argued our position on copyrightability. Our view of 102 is that it's a safe harbor. Your Honor's view is a different view and your Honor has rejected our view. But we understand here that what we are asking the Court to do is precisely what the Court wants to do, that we are asking the Court to weigh and balance all appropriate considerations in the Court's view and do what the Court considers to be the fair thing to do and in the public interest. As the case law has said, fair use is an equitable rule of reason. It's used as a safety valve. Once a court concludes that the defendant has committed a copyright violation, has copied copyrightable subject matter, the court in its discretion using fair use can hold that that use is privileged, that it's not a violation of the copyright law.

Now, the statute lists four factors to be decided by the court in analyzing fair use and I'll discuss each one very briefly in a moment, but the point I want to make is that the statute says that these factors aren't exhaustive—I beg your pardon—that the case law and the legislative history says that these factors are not **[3-45]** exhaustive. According to the *Galoob* case they are intended to guide but not to limit analysis, and the *Galoob* case is relying on the legislative history that is referenced in that case. It is clear that the statute doesn't say how much weight you're to give to each of the factors or indeed to any factors that your Honor might consider. That's up to the Court's discretion.

Now, your Honor, I would like to briefly run through the four factors. What *Sega* says is that the second factor is to be considered the most important in a computer case of this type. *Sega* says that at page 1522, but I'd like to start with, of the four factors, the factors that Lotus might argue go more strongly to its side.

And the first factor listed is the purpose and character of the use, and that's the discussion of whether the use is commercial in part or in whole or how much and so forth. And the case law is very clear that copying for a commercial purpose weighs against finding of a fair use, but the case law is also clear that that can be rebutted by characteristics of a particular commercial use. All the cases that we've cited say that. *Sega* says explicitly the commercial nature of a use is a matter of degree and not an absolute. The *Galoob* case says that when evaluating the nature of the use as to whether it's a commercial use, the court should look first to the user's use of the Lotus menus **[3-46]** in the Borland product. That's where you start, the user's use. And we would argue and we believe the record shows and we would cite to your Honor's prior opinions that Lotus encourages people to write macros and freely admits the users own those macros, and Borland is giving people an avenue to run their macros, and that's the nature of the analysis that one might undertake. *Sega* at page 1522 indicates that it is important to note in a case like this that there's no evidence that the defendant sought to avoid performing its own creative work, and we would respectfully

argue to your Honor that we would make that same argument here. And finally with respect to the first commercial use factor, the case law says that the Court is free to consider the public benefit resulting from a particular use notwithstanding the fact that the alleged infringer may gain commercially. So that's the first factor that Lotus might suggest cuts in its favor.

The second factor that Lotus might say cuts in its favor is the fourth factor in the statute, the effect of the use upon potential market for or value of the copyrighted work. And the case law here is very clear. It says the fourth factor looks primarily to whether a use supplants—and that's the quotation—supplants the normal market for the copyrighted work. The *Galoob* case says, quote: "A fair use will frequently suppress demand **[3-47]** for a work, but as long as it does so without supplanting demand, the indirect detrimental effect on the market is not the subject of copyright protection." *Sega* is to the same effect, your Honor. It says: "While a use that effectively usurps the market for the copyrighted work by supplanting that work may be dispositive, quote, the same consequences do not and could not attach to a use which simply enables the copier to enter the market for works of the same type as the copied work."

Now, your Honor, we bear the burden here and I believe that on the record that will be tendered to your Honor we meet that burden, and we rely primarily on one document and it is the document that your Honor has heard so much about yesterday. It's the Wall Street Journal ad that your Honor will hear further argument about. This ad was published in both the Wall Street Journal and the New York Times. It's found in Volume IV, tab B-5 of the authorities in front of you. It was published by Lotus in those newspapers shortly after your Honor's July 31st opinion. It says a lot of things which are relevant in this case, particularly to the issues of macros and macro compatibility, but on the fair use point it says they've lost in the marketplace where 1-2-3 for DOS continues to dominate. "Based on a recent report from the Software Publishers Association, Lotus 1-2-3 accounts for seven out

[3-48] of ten new DOS spreadsheets and over 80 percent of all DOS spreadsheet revenue." So, your Honor, we would respectfully argue that we have not supplanted the demand for Lotus products in any sense. Now let me move to the other fair use factors in the statute.

The second factor is the one that the *Sega* case says should be considered by courts in copyright cases, and that is the nature of the copyrighted work. And *Sega* comments extensively on the fact that when a fair use case deals with a computer program, because of the nature of computer programs, fair use of the expression involved is to be easily found.

Now, I know your Honor is familiar with this case and I don't want to go through reading the quotations from it, but the argument that I do want to make is that the arguments that we have made to the Court and that the Court has rejected in the context of 102, the arguments about *Baker v. Selden* and *Feist* and *Altai*, *Sega* says are dispositive or relevant, at least, on the issue of fair use. And we would respectfully ask your Honor to consider those arguments in evaluating fair use here and consider the nature of the work, the utilitarian nature, whatever expressive qualities your Honor thinks it has, and consider that in the context in which *Sega* suggests that fair use of computer code expression in certain circumstances under **[3-49]** those cases is permissible.

The third factor and the last one I'll discuss from the statute is the amount and substantiality of the portion used in relation to the copyrighted work as a whole. According to *Sega* at page 1526, if Borland copied the entire work at issue, that would not preclude a finding of fair use.

The point I want to make about this, your Honor, is, we copied only as much as was necessary to give us macro capability. Your Honor has said in its prior opinions that the look of these products is very different. No one looking at them confused one from the other. The fanciful aspects that we previously discussed, the pull-down menu motif, the use of icons, on-line help, things like that, everyone admits are different, and we would suggest, therefore, that the third factor, like the second factor, cuts heavily in our favor.

Finally, your Honor, I would like to respectfully argue to the Court—and again I say that we understand we're addressing the Court's discretion here, we understand the case law that says that fair use is an equitable rule of reason, but I'd like to make a couple of other observations which I would respectfully ask the Court to consider in the context of fair use.

I would like the Court to consider the fact that **[3-50]** reasonable minds might disagree on the proposition that the menu command hierarchy is copyrightable. Certainly that was the case in 1987 when Borland did what it did, and I would respectfully ask your Honor to weigh that in the balancing of what's appropriate in terms of relief.

I would also ask your Honor to consider the effect of the relief upon Borland and upon the market generally. If Lotus asks for monetary recovery, money would be taken from Borland that would be used for software development and transferred to Lotus. I would respectfully argue that in a situation like this, I would ask that the Court's equitable powers not be used in that way.

The same is true of injunction, your Honor. I would argue that an injunction is not appropriate or necessary here given the balancing of fair use.

The point I want to make is that we understand from your Honor's prior decisions that your Honor considers aspects of the menu command hierarchy to be expressive. I am not now quarreling with that. Your Honor also considers aspects of the menu command hierarchy to be functional and utilitarian. I am arguing that we use the functional and utilitarian aspects and I would argue to your Honor and ask your Honor to consider that that is a fair use.

Let me now turn in the time remaining that I have to the issue of functional constraints. As I say, your **[3-51]** Honor, if I don't get through that fully, that is found and briefed in considerable detail at page 41 of the brief that we have submitted.

The question put to us by the Court in its procedural order at pages 20 and 21 is, quote: "To what extent, if any, did functional constraints limit the number of possible ways in which the Lotus 1-2-3 menu command hierarchy could have been arranged at the time of its creation." That's the question put to us. I do think, your Honor, that for the most part this question will be reduced to a legal argument on both sides.

Let me say that I believe it is uncontested in the record they have proffered some of Mr. Kapor's declaration from the *Paperback* case. We will be proferring the remainder of it. I think it is clear that Mr. Kapor chose the words in the menu command hierarchy according to functional rules—I would cite Kapor Affidavit, paragraph 75, for that proposition—rules like the word has to intelligently convey to the user the purpose of each command, rules like a word on a particular menu level has to begin with a unique letter and so forth. We cite in our brief and we will tender the testimony of Ezra Gottheil, who was Lotus' 30(b)(6) designee, to the same effect. And we tender also testimony of Lotus' expert, Bernard Galler, to the same effect. So that's the nature of the record with **[3-52]** respect to the selection of the words.

We think that the record is also clear based on Mr. Kapor himself that the organization of those words was dictated or governed or implicated, whatever the appropriate word is, by five functional rules. And they are set forth in the brief and I won't read them, but they are set forth at Kapor paragraph 76, 77 and 78. They are things like only seven plus or minus two items on the line, words most frequently to the left and the top, that kind of thing. It is also clear in the record, the record that we'll be tendering and in fact Lotus has tendered, that the rules apply to the entire hierarchy, to the top level and to subsequent levels. Mr. Gottheil says that. We have also introduced some testimony. And the Court has said in its *Paperback* decision that the words were ordered according to predicted frequency of keystrokes.

Now, in prior treatment of this issue the Court had asked the parties—it was not clear to us and we asked the Court exactly how the Court wanted the issue framed. In other words, if the question were is the menu command hierarchy guided by functional principles, we think that the answer to

that is clear. The answer is yes. If the question is to what extent—do the application of these rules mean that invariably and inalterably there would be only one result, then the answer is almost certainly no. [3-53] And the Court said, "I understand that. I understand that there's something between one and a million and a half different ways to structure this. About where is it on the chart?"

Now, I guess our first attempt at answering that, which is not really an answer, is to say that under the law we don't think that is the right question, although certainly it's the right question in terms of the one that we're called upon to answer and I'll try to do that. But our point is that because the work was prepared according to functional rules as was the directory in *Feist*, for example, that makes it uncopyrightable whether there's one way to do it or 406 ways to do it. Insofar as we are able to determine, I cannot come up with a numerical result based on this record as to the exact number of ways that this can be done. *I* think that the Court can look at the record and try to make that determination itself, and if the Court wishes to do that, I would point out some of the testimony which we will be tendering and some of which has gone in for the proposition that these functional rules seriously constrict the number of choices for each word and its placement.

And I would cite first to the testimony of Dr. David Liddle at page 121. He is our expert. Your Honor asked yesterday how his name was pronounced and I'd just like to take a minute to explain who Dr. David Liddle is.

* * *

[3-58] not the question that your Honor has asked us to decide with respect to our closings.

And with that, your Honor, I'll sit down. Thank you, sir, and I appreciate the opportunity to address the Court.

THE COURT: Thank you.

MR. GUTMAN: May it please the Court, I had originally planned today to discuss three issues, but with the Court's

permission I think I'd like to start with a fourth issue which was not on my agenda but obviously was on Mr. Reback's and I would like to deal with it briefly, and that's the question of fair use.

Your Honor heard me rest before this argument. Because of the accommodation we've granted Mr. Reback we've not heard him say those words, but I'm counting on the Court to keep him from in any way expanding or altering his evidentiary submissions on this case based on what he's hearing now and wouldn't normally hear until after he rested. But there's a funny thing about the fair use defense. I believe it's fair to say that it's an affirmative defense. I believe they have so treated it in their answer to the supplemental complaint, but it was nowhere in their answer to the original complaint that they filed a couple of years ago. I would submit that with respect to the issues we are addressing today and that we [3-59] have been trying this week, they have waived a fair use defense. They did not raise it in a timely fashion and I would suggest to your Honor that closing argument is a little bit late.

Second, again, unless the Court were to give them leave after what I'm about to say to expand the record, they've made no showing—even if all of my objections to their evidence are overruled this afternoon, they haven't come close to making a showing of fair use on this record. I've read *Sega* and I've read *Galoob*. Neither helps them, your Honor.

In *Sega* the issue was whether intermediate copying akin to research; that is, the taking apart, the decompilation of a program, when according to the Ninth Circuit you could establish that this was the only way that you could get to the unprotected ideas, could under appropriate circumstances be a fair use. Now, that may or may not be a right decision. It's a very controversial decision. It has not been uniformly followed across the country. A similar case in the federal circuit involving similarly related companies where the federal circuit was applying Ninth Circuit law I'm not sure is entirely consistent with it, but even if we assume that *Sega* were right, it doesn't help them here. It's got nothing to do with this. The *Sega* court was quite clear that what it was [3-60] distin-

guishing there was between the copying akin to going to a library and taking notes from a book and then not using those notes in the final product. The court gave the defendant no free pass if the final product they sold was copied. The limited fair use exception recognized in *Sega* was to the extent that it was the only way to get access to the unprotected ideas. If the only way to do that was to make an intermediate copy; that is, to decompile so that you could get a reconstruction of the source code, the Ninth Circuit said that was okay. So to get from there to here is a tremendous jump.

They have *Galoob*. In the *Galoob* case the issue there was the game genie which speeds up the Nintendo video game system, your Honor, and one of the issues there was the issue, the same issue or issues very close to the *Sony Betamax* case in the Supreme Court where one of the questions was can the person at home using one of these game genies be found to have infringed the copyright, and the court said no, that's a fair use. Among other things, it's not a commercial use. That is a far, far cry from where we are here.

Section 107 of the act lays out the factors. It is correct as the supreme Court said in *Harper & Row* that those factors are not exclusive. I have certainly not put in what proof I would have put in on that issue in this case **[3-61]** had that issue been properly in this case. But if we take Mr. Reback as having done so, your Honor, and if he has constructively rested on his record as it exists subject only to my ability to shoot it down based on my objections, and even if the Court holds that he didn't waive this defense by failing to include it in his original answer, he has still failed to meet the 107 factors.

I mean, looking at the one exhibit, the advertisement which he says is the heart of his case on fair use, that doesn't say that they haven't caused us serious commercial harm. All it says is they haven't killed us. They have caused serious commercial harm. And notwithstanding Mr. Reback's appeal to the equitable concerns of the Court and please don't deprive Borland of money to do research in software development, although Lotus could make better use of the money, I would

suggest, in terms of its own development efforts, we will show when we get to the remedy phase, we will show—and that will be a jury trial on damages—that they have injured us to the tune—assuming that your Honor accepts that my theories of damages is acceptable for the jury, and assuming that the jury agrees with what case we're going to put on, we're going to show that they have damaged us to the tune of hundreds of millions of dollars, and that will easily—I haven't put that proof in now, but that would easily blow **[3-62]** away whatever inference they want to draw from that one advertisement concerning whether or not they've had a serious commercial impact on my client's activity, and at the appropriate phase of this case we're absolutely prepared to prove that and we're looking forward to the opportunity.

Now, again, reserving my right to address fair use in a more substantial and serious fashion and with supporting evidence, assuming that I'm incorrect in the assumption that that issue has been waived for the issues that we're dealing with in this phase of the case, I'd like to move on if I could to the other issues that Mr. Reback raised and one he didn't.

I'd like to start with one he didn't. It's the question of copying, your Honor. This summer your Honor ruled on summary judgment and held as a matter of undisputed fact that Borland had copied the menu commands, menu command structure, keystroke sequences and macro language of Lotus 1-2-3 in the Quattro and Quattro Pro program. That ruling was correct. The material facts are undisputed. We are content with that record.

However, at the January 14th hearing your Honor asked us a question, and this is what Mr. Reback refers to as the fair use question. I understood fair use to be a fairly small part of the question. And the question your Honor put was, would it make a difference—if in fact the

* * *

[3-92] Friday, I would like to make sure that any of our completeness objections, any of our Rule 32 objections, are included. So with the caveat of making sure that there are not

any counter-designations to Digate that ought to be in there for fairness, it is correct that on their affirmative defense, excluding Key Reader which we are not dealing with today, our objections are to the Declaration of Robert Kohn as specified and to four documents that were appended to the Declaration of Edmond Gregorian.

MR. DETKIN: Your Honor, I would take exception to any further counter-designation in that if anything was clear from the January 14th transcript, it was that the parties were standing on the record as it existed at the time of summary judgment. That's precisely what was submitted, it's my understanding that was what Lotus has relied upon in their designations, and that's what we're standing here prepared to argue.

MR. GUTMAN: As your Honor observed earlier on summary judgment, we did not go through evidentiary objections, including completeness. If he would like me instead to object to Digate, if we had a few moments to review their designations, I could probably come up with objections and we could keep what they want to put in of Digate out. What I was proposing here was that we instead treat these deposition transcripts the way we have most of **[3-93]** the others in which most of the objections were waived provided that completeness counter-designations all went in. I'm happy to do it either way, but I certainly did not waive my right to object to Digate and I don't believe I waived my right to object on grounds of completeness and to offer counter-designations.

MR. DETKIN: Your Honor, this may be a hypothetical argument. I'm willing to allow Mr. Gutman to take the time to look and see if he wants counter-designations, to see if we object to the counters—

THE COURT: All right. Well, then I will receive in evidence all of the matters proffered by Borland as listed in the citations to the record by both parties, the second page in in docket number 312, subject to my hearing and ruling on the objections to the Declaration of Kohn and to my hearing whatever there is to be heard with respect to the Digate designations. All right.

MR. GUTMAN: And, your Honor, there were also—I think I mentioned this a moment ago. There are also four documents on that list that we've objected to, they were exhibits to the Gregorian declaration, but with that addition that's where we understand us to be.

MR. DETKIN: That's fine, your Honor.

THE COURT: All right. Excuse me just a moment.

[3-94] MR. DETKIN: Your Honor, at this time we would offer the Declaration of Robert Kohn.

THE COURT: Wait just a moment.

(Discussion off the record with the clerk.)

THE COURT: The clerk is advising me that if we're going to have to have carry-over time tomorrow, she needs to know immediately to get it on the schedule. I think I'll just forget about that, give her the message not to worry about it. I trust we'll be able to finish today. If we aren't, the published schedule will be a little different from what I do tomorrow, but we'll live with that.

All right. Go ahead.

MR. DETKIN: At this point, your Honor, I would offer into evidence the Declaration of Robert Kohn.

THE COURT: All right.

MR. GUTMAN: Does your Honor have handy a list of our objections, plaintiff's objections?

THE COURT: I think so, but now let me just—

MR. GUTMAN: This was the addition on affirmative defenses that was filed this morning headed "Plaintiff's Objections to Borland's Record Designations Concerning its Phase I Affirmative Defenses."

THE COURT: Yes, I have it before me.

MR. GUTMAN: Our first objection, then, comes on paragraph 4, second sentence, on the grounds stated.

[3-95] THE COURT: Wait just a moment. Let me get it before me.

(Pause.)

THE COURT: And what are the grounds of objection?

MR. GUTMAN: Well, basically two, your Honor. To the extent that the sentence is using the royal "we," it's hearsay.

I don't believe Mr. Kohn is in a position to testify to what other people were aware of.

The second objection is, to the extent that it purports to discuss the content of various other statements, public statements by Lotus representatives, then there's another level of hearsay in here. So depending on the use for which this sentence is offered, I think we have a valid objection.

MR. DETKIN: Your Honor, as to the first objection, the royal "we" objection, I would point out that as noted in the first paragraph of Mr. Kohn's declaration, he was vice-president of the company, he's the secretary of the company, he's also general counsel, he was authorized to speak on behalf of Borland, and I believe there should be no problem with using the "we" to refer to Borland.

MR. GUTMAN: That may make everything he says an admission of Borland, but it does not give him the right to say what everybody at Borland thought or were aware of, **[3-96]** et cetera.

MR. DETKIN: As to the second objection—

THE COURT: Well, wait a minute. I think I should rule as follows on that objection, that I receive this as evidence of Kohn's awareness. If it becomes material to consider whether other persons were aware, other specific persons were aware, I cannot receive it as evidence that every other person in the Borland hierarchy was aware and therefore I can't receive it as to evidence that some other particular person was aware. So I sustain the objection to that limited extent, but I overrule the objection to the extent that I will consider it as evidence of Kohn's awareness.

MR. GUTMAN: Your Honor, I may not have stated it crisply enough, but there are really two objections.

THE COURT: Well, now, I'm coming to the other point.

MR. GUTMAN: Thank you.

THE COURT: The other point is that this does not identify what public statements are being referred to.

MR. DETKIN: Your Honor, that's then in paragraph 5. The next paragraph discusses that.

MR. GUTMAN: And the second and third sentences we've objected to as hearsay.

MR. DETKIN: Your Honor, we don't offer this **[3-97]** for the truth of the matter asserted.

MR. GUTMAN: I think they do. If not, I have a relevance objection.

THE COURT: All right. If you're offering it as evidence that statements were made by a Lotus executive, somebody who spoke with authority for Lotus, when the only evidence you have that such a statement was made by somebody with authority for Lotus is a report prepared by some other person purporting to say what some Lotus executive said to that person, then we do have a hearsay problem that I don't think is solved. So it does seem to me—I had understood that you were offering it for the truth of the assertion that a Lotus executive said this.

MR. DETKIN: We are offering it for the truth of the assertion that Mr. Kohn had read these articles in which the Lotus executive was quoted as saying this.

THE COURT: But not for the truth of the assertion that a Lotus executive did say this.

MR. DETKIN: That's correct, your Honor.

THE COURT: Well, all right. I receive it for that limited purpose, but I tell you that that doesn't mean that you have evidence that a Lotus executive actually said that.

MR. DETKIN: I understand.

THE COURT: All right.

[3-98] MR. GUTMAN: In fact, that witness testified that he never said it.

THE COURT: All right.

MR. GUTMAN: Your Honor, the same objection, since we've moved to our paragraph 5 objections, covered not just the second and third sentences of paragraph 5 but Exhibit A as well, which is the article.

THE COURT: Yes. I take it the statement you made refers to that as well and the purpose for which it's being offered.

MR. DETKIN: That's correct.

THE COURT: All right. The ruling I've made applies to that also.

MR. GUTMAN: Okay. Your Honor, as to the fourth sentence of paragraph 5, the same objection plus—the same types of objections, but not so easily cured. Because as we state in the document, we don't think that these passages would be admissible even as to the witness's state of mind, because there is no evidence in the record that he saw them in a timely fashion. And indeed, if one looks at the exhibits themselves, they bear the stamp "Lotus Library" and our production numbers, which makes it clear that these particular copies of these articles were the copies that we produced to them in discovery which by definition Mr. Kohn could not have had in front of him and **[3-99]** read at that time.

MR. DETKIN: Your Honor, that does not prove that he didn't use them at the time they were published—

THE COURT: Well, but he doesn't say that he did, so what evidence do I have before me—how can I receive this as evidence that he did read them when he doesn't even say so?

MR. DETKIN: Well, your Honor, I think the fair inference of paragraphs 4 and 5 together is that he did read them; however, that only goes to the relevance and not the admissibility of this particular evidence.

THE COURT: Well, it's not relevant unless it's admissible, and it's not admissible unless there's some evidence before me that some Borland executive read it, and this is not a statement that he read it. And I have to say to you that I assume the affidavit is carefully drafted and if he'd read it he would have said so, and when he doesn't say he read it, I don't think it's appropriate for me to draw the inference by looking at the two paragraphs together that he read it. So I sustain the objection to the fourth sentence of paragraph 5, and I take it that applies to Exhibit B as well.

MR. GUTMAN: Yes, your Honor.

MR. DETKIN: Your Honor, paragraph 4 does say that he was aware of public statements—that's plural—**[3-100]** of Lotus representatives, and that is a lead-in to paragraph 5 in which

he is then describing the statements. Again, I believe the fair inference—it may not have been drafted—

THE COURT: I'm sorry. I don't think I can properly read it that way. When things have been prepared as carefully as they have been for this trial, that kind of failure to state the critical question that bears upon admissibility it seems to me I should not infer was an accidental omission. I sustain the objection to the fourth sentence of paragraph 5.

MR. GUTMAN: The next objection, your Honor, is paragraph 11, first sentence.

MR. DETKIN: Your Honor, actually, the next three objections they have, they admit in writing that those paragraphs are admissible for state of mind, which is the purpose for which they're offered.

MR. GUTMAN: As long as that's the limited purpose, since he had no firsthand knowledge.

THE COURT: So the paragraph 11, first sentence; paragraph 12, first sentence and Exhibit C; and paragraph 13, second sentence, are offered and received only as to the witness's state of mind.

MR. DETKIN: That's correct, your Honor.

MR. GUTMAN: The next one, your Honor, is **[3-101]** paragraph 14.

MR. DETKIN: Actually, your Honor, I'm sorry. On paragraph 13—I'm sorry. I went a little ahead. Paragraph 13 I believe he can speak to of his own personal knowledge. That paragraph refers to an announcement of the Quattro product, the Borland product, as you know, and it's a statement that Borland's announcement received first-page coverage in the trade press. Mr. Kohn is an executive in a software company. I certainly think that he is qualified to state what did and did not appear in the trade press.

MR. GUTMAN: Your Honor, we don't know where, we don't know when, we don't know which publications.

MR. DETKIN: That's relevance. That's not admissibility.

THE COURT: Well, how is it going to help me if I don't know the answer to those questions somewhere in the record? How can I use it in my fact-finding?

MR. DETKIN: Your Honor, the point to be made by this paragraph I don't believe was so critical that it required a listing of the various trade press in which this announcement appeared.

THE COURT: Well, all right. Now, I will overrule the objection. I'll tell you that the lack of specificity has a bearing on what weight I can give to paragraph 13, but I will overrule the objection.

[3-102] MR. GUTMAN: Just so that your Honor understands the state of the related record, neither party has ever found or submitted any such article.

THE COURT: All right.

MR. GUTMAN: I would like to note that omission.

THE COURT: All right.

MR. GUTMAN: And so far as we know, there isn't one.

The next is paragraph 14, second, third and fourth sentences. Again, your Honor, we've got Mr. Kohn speaking on behalf of the corporate entity, we've got the royal "we," we've got speculation as to the thoughts and mental processes of other individuals.

MR. DETKIN: Your Honor, we have no objection if you treat this so-called royal "we" in the same manner you treat the others.

MR. GUTMAN: If they're offering it only for his state of mind and the Court believes it would be relevant on that basis.

MR. DETKIN: Well, as to the royal "we," whether you're talking about Kohn's state of mind or others in the company, I do agree with the ruling your Honor set forth before and I think that's a proper handling. As for the remaining hearsay, I believe he is testifying here as to **[3-103]** facts, as to what he believed, and he of course is competent—

THE COURT: But that's making it relevant only to his state of mind.

MR. DETKIN: Fair enough, your Honor. That's fine.

THE COURT: All right. So I think that sums up, that I receive the second, third and fourth sentences of paragraph 14 in relation to his state of mind, not for any other purpose.

MR. DETKIN: That's fine.

THE COURT: All right.

MR. GUTMAN: Paragraph 15, your Honor, on the grounds stated. I mean, this is even more extreme. Here he's purporting to speak on behalf of everyone at Borland, "Borland having heard nothing from Lotus." He has no basis for that.

MR. DETKIN: Well, first I note that they won't deny that Lotus never said anything to Borland, but getting beyond that, he's clearly competent to testify as to what was or was not received from Lotus. He's the general counsel, had—

MR. GUTMAN: He would not be in a position to speak to every possible communication between anyone at Lotus and anyone at Borland, and if he did it would be **[3-104]** hearsay. If this were limited to his—

THE COURT: I again—you know, there's another ambiguity about this "having heard nothing from Lotus." In the preceding paragraphs we've been talking about things that were heard directly and indirectly from Lotus, including alleged public statements. I think that insofar as that phrase "having heard nothing from Lotus" is concerned, I should receive it as evidence that this witness had heard nothing from Lotus other than what he's been saying he's been hearing indirectly. I take it this means no direct communications from Lotus.

MR. DETKIN: That's fine, your Honor.

THE COURT: I receive it for that purpose.

MR. GUTMAN: Paragraph 16, second sentence.

MR. DETKIN: Your Honor, we only offer this as is stated, permissible to show Mr. Kohn's state of mind.

THE COURT: Just a moment. (Pause.)

THE COURT: All right. I receive it for that limited purpose. I take it you're not pressing the objection against its being received for that limited purpose.

MR. GUTMAN: No, your Honor, only his state of mind for whatever relevance that has.

THE COURT: All right.

[3-105] MR. GUTMAN: Paragraph 17, second sentence, lack of personal knowledge.

MR. DETKIN: Your Honor, we would only offer that to show his state of mind as to what he believed and his belief in the ownership of "Lotus Magazine."

THE COURT: Well, all right. I receive it for that limited purpose.

MR. GUTMAN: Paragraph 18, third sentence.

MR. DETKIN: I believe this is the royal "we" objection again and we have no objection to it being treated as your Honor has treated the other—

(Defense counsel confer.)

MR. GUTMAN: Your Honor, I'm sorry. The second sentence has the problem too. Well, I guess it says "to the best of my knowledge," which means "so far as I know."

THE COURT: So you're proposing that I receive it only as to Kohn's state of mind?

MR. DETKIN: As my partner points out, we do believe he can speak for the company in that he is the corporate counsel, general counsel, and a vice-president and officer of the company, but, you know, that was your ruling before and we're willing to live with that ruling.

THE COURT: All right. He may or may not speak for the company on a particular issue. We've got the **[3-106]** problem of entity state of mind and we have to be concerned about who speaks for the entity on such a matter. If that becomes material, it would become an issue perhaps of substantive findings rather than admissibility of evidence. So if you are offering this only as to Kohn's state of mind, I think I should receive it for that limited purpose.

MR. DETKIN: We would be happy to say that at least for that purpose. We believe it is appropriate for other purposes, but we understood the Court's ruling on that matter.

THE COURT: All right. I receive it only as to Kohn's state of mind.

MR. GUTMAN: Thank you, your Honor. Paragraph 20, third sentence. This sentence confesses, your Honor. It begins: "Although I cannot state from personal knowledge."

MR. DETKIN: Well, I believe that's reading it out of context. He's referring to whether a common practice of Borland,

which he is competent to testify, was followed on a particular occasion.

MR. GUTMAN: I would beg to differ, your Honor. It says: "It is common practice for representatives of our competitors." This is—

THE COURT: Well, I receive it and will consider it literally. It doesn't say that's a common **[3-107]** practice of Lotus. It does speak to a common practice of representatives of competitors. I will receive it literally.

MR. GUTMAN: That's fine, your Honor.

THE COURT: Now, excuse me for interrupting this process, but I will have to break at this point and we'll break until 2 o'clock.

MR. GUTMAN: Thank you, your Honor.

MR. DETKIN: Thank you, your Honor.

(Lunch recess taken at 12:43 p.m.)

* * *

(2:00 p.m.)

IN OPEN COURT

THE CLERK: Court is now in session. You may be seated.

MR. GUTMAN: Good afternoon, your Honor.

THE COURT: Good afternoon.

MR. GUTMAN: I believe when we broke for lunch we were up to—we had just completed our objection to paragraph 20 of the Kohn declaration, and the next one is paragraph 22 where our objection is as stated.

THE COURT: All right.

MR. DETKIN: Your Honor, this is a relevance objection I believe that Mr. Reback explained this morning about Borland's acquisition of Surpass.

[3-108] THE COURT: All right. I'll take that with the case.

MR. GUTMAN: I think there may be two separate relevance points here, your Honor, but I understand that your Honor is taking it with the case.

THE COURT: Well, it was also listed on—no, wait a minute. That was 22. I'm not sure I understand what you're saying to me. Two separate relevance objections?

MR. GUTMAN: Well, yes. Your Honor, to the extent that all they are trying to say in paragraph 22 is that they had discussions concerning a possible acquisition of Surpass, subject to there being something else in the record that links it up along the lines of Mr. Reback's argument this morning, and I'm not sure there is, that would be relevance objection number one. That is, if there isn't anything to link that up, then we would have an objection on relevance.

The second is, to the extent that they are intending to imply something about Lotus because of the references to Surpass' 1-2-3-compatible spreadsheet, that I think is a different point, and again, we would just want to preserve a relevance objection there.

THE COURT: All right.

MR. GUTMAN: Paragraph 23 has the second of those two relevance objections, namely the relevance of **[3-109]** Lotus' behavior toward Surpass coupled with the further problem of hearsay, because it purports to report what some out-of-court person said to Mr. Kohn.

MR. DETKIN: Your Honor, that's offered to describe Mr. Kohn's state of mind, or his diligence process with respect to the Surpass acquisition.

MR. GUTMAN: If it's clear that they are not offering any of this as any kind of evidence anywhere as to what Lotus ever did or didn't do with respect to Surpass, if that's how I understand it, then I guess there's still a relevance objection, but we've at least cleared up the purpose of the offer. But if they intend to argue—

THE COURT: Well, I cannot take it as evidence that Surpass never received any notice claim by Lotus, so that subject is plainly hearsay because it's a declaration purported to be a declaration of some person not here present.

MR. DETKIN: I understand that, your Honor. I believe there's plenty of other evidence in the record with respect to that matter.

THE COURT: Well, all right. Then it's received only for Kohn's state of mind.

MR. GUTMAN: Paragraph 24, your Honor, for the reasons stated.

MR. DETKIN: The relevance objection?

[3-110] MR. GUTMAN: Well, plus the others. It's not simply relevance.

MR. DETKIN: I'm not sure I understand the objection, your Honor.

THE COURT: Well, I understand the objection to be that if you are offering this to support an inference by me as fact finder that Lotus knew of the Surpass acquisition or knew of the use of the Surpass technology in Quattro Pro, then it's not admissible for that purpose.

MR. DETKIN: Your Honor, we're not offering it for that point. We're offering it for the point which is stated here, which is that Borland did in fact incorporate the Surpass technology into Quattro Pro and during the 18-month period between the time it acquired Surpass and shipped Quattro Pro did not receive any notice from Lotus. That is all it's offered for. There's no other inference intended.

THE COURT: All right. Received for that limited purpose.

MR. GUTMAN: Your Honor, there was one further objection in our list, lack of personal knowledge concerning whether or not Borland received notice. I mean, if that's limited to the personal knowledge of this witness—

MR. DETKIN: Well, your Honor, again, we maintain that a corporate officer can speak on behalf of **[3-111]** Borland. I understand you've overruled me on that.

THE COURT: Well, that's correct. He can speak for himself, but the fact that one corporate officer knows it doesn't mean that every other corporate officer knows it.

MR. DETKIN: I understand that, your Honor. Our position is that he can speak on behalf of Borland, I understand you've overruled me on that, and we're willing to allow you to—

THE COURT: All right. When one has authority to speak on behalf of an entity for one purpose, it doesn't mean he speaks on behalf of the entity for every other possible purpose.

All right. Paragraph 25.

MR. GUTMAN: Relevance and lack of personal knowledge, your Honor.

MR. DETKIN: The relevance, I believe, is explained in our papers. Lack of personal knowledge, I'm not sure I understand.

MR. GUTMAN: How would Mr. Kohn know everything that was or wasn't said in communications to which he was not a party, or how would he have firsthand knowledge indeed of the frequency of those communications or even that they took place?

THE COURT: He cannot speak for anyone other [3-112] than himself on these subjects.

MR. DETKIN: I understand that, your Honor.

THE COURT: All right. I receive it for that limited purpose.

MR. GUTMAN: Paragraph 26 and Exhibit E, your Honor, relevance for the whole thing, and also as to the second and third sentences, hearsay and lack of personal knowledge.

MR. DETKIN: I believe the basis for the knowledge is set forth in this paragraph. He attaches the trade press to which he refers for the release of SuperCalc5.

THE COURT: Well, what is Exhibit E? Is it attached here?

MR. DETKIN: It should be, your Honor. It's an article dated June 20th, 1988 from a trade press called "InfoWorld."

MR. GUTMAN: The paragraph does not say, your Honor, that he saw that article at the time, and the article itself is hearsay if they're offering it for the truth of any of its contents, including any of these descriptions of the menu, macro and file compatibility.

MR. DETKIN: The product itself is in the record.

MR. GUTMAN: No, it's not.

[3-113] MR. DETKIN: Is that 4 or 5? I apologize, your Honor. That was a different version of the same product. Mr. Kohn can certainly testify as to the features of SuperCalc and certainly as to the features of Quattro and Quattro Pro, which is the third sentence to which he refers. The actual release date—

THE COURT: Well, I have some problem with that. That is, when he is drawing comparisons between products, that's not a recitation of factual observations. That's an expression of his conclusions.

MR. DETKIN: Your Honor, according to this paragraph—

THE COURT: Does this witness purport to be capable himself of drawing the comparisons that are being made?

MR. DETKIN: He is an officer of a spreadsheet company, a software company, and—

THE COURT: Are you telling me that every officer of a spreadsheet company can do source code and object code himself?

MR. DETKIN: Your Honor, this is the features. This is stuff that comes up on the screen.

THE COURT: Well, I know, but I'm trying to get at the premise of your assumption that every officer of a software company knows everything there is to know about [3-114] software. That plainly is not so, is it?

MR. DETKIN: No, your Honor, I'm not intimating that it is. However, what is clearly inferred here is that he looked at the products. He is certainly capable of loading these products on the machine and looking at what features that SuperCalc has and he certainly knows what features Quattro and Quattro Pro have.

MR. GUTMAN: He never says he looked—

THE COURT: You know, I have another view that I frequently express in this courtroom with respect to expert testimony.

I do not credit any expert opinion that cannot be explained to me in a basis that I can understand it as a reasoned opinion. I think that's what I should do. I shouldn't be in the position of deciding that I should credit one expert's testimony rather than another expert's testimony on the basis of their respective qualifications or their credibility as they appear before me or whatever. If I am expected to receive and rely upon an opinion from an expert, I insist that it be explained to me in a way that I can understand it. That's the only way I can assess the validity, the credibility of the opinions themselves, and there's no way I can get that kind of help out of this kind of conclusion.

So I think what I should do is say I'll overrule [3-115] the objection, but I tell you how I go about evaluating testimony

that's placed before me when it's placed before me in opinion and conclusion form without any explanation that I can understand and assess, evaluate.

MR. DETKIN: That's fair enough, your Honor.

THE COURT: All right.

MR. GUTMAN: Your Honor, just so the record is complete, if in fact they are offering this as expert testimony, we would add the objection that Mr. Kohn, who is only a lawyer, would not be qualified as an expert in this field.

THE COURT: Well, I understand that. I'm saying whether it's expert opinion or just conclusions, that we sometimes let lay witnesses—in either event, when I'm trying to understand something as fact finder and make an evaluation, I weigh opinions and conclusions when I think they're properly before me on the basis of being able to understand them and understand the reasoned basis for it, and if that hasn't been provided for me, it doesn't carry anything like the weight that it would if it is provided for me.

MR. GUTMAN: Thank you, your Honor. The next objection is to paragraph 27 and Exhibit F on the grounds stated.

MR. DETKIN: Your Honor, again, this goes to **[3-116]** Mr. Kohn's state of mind in his capacity as general counsel and secretary of the corporation upon reading the policy statement which is reported.

THE COURT: Well, if that's the sole purpose for which it's offered, I'll receive it as bearing on his state of mind.

MR. GUTMAN: I would just add for the record again that what he claims to have read—actually, he doesn't claim to have read the policy statement and I'm not even sure he claims to have read Exhibit F, but Exhibit F, just so we're clear, is not the statement itself. It's a B & A report concerning the statement.

THE COURT: All right.

MR. GUTMAN: Paragraph 28 and Exhibit H.

MR. DETKIN: Your Honor, I believe their objections go to the weight of the evidence and not their admissibility.

MR. GUTMAN: There's no evidence here that he had personal knowledge of these things, your Honor.

THE COURT: Now, again, if you're offering it as bearing on his state of mind, it's one thing. If you're asking me to receive for a broader purpose than that a statement such as that first sentence, then it's not appropriate.

MR. DETKIN: The first sentence, I believe, **[3-117]** does go to Mr. Kohn's state of mind as—

THE COURT: All right.

MR. DETKIN: As to his personal knowledge, paragraph 1 sets forth clearly he has personal knowledge.

THE COURT: I'm sorry. Wait a minute. I didn't understand that. What sets forth personal knowledge?

MR. DETKIN: Paragraph 1 says that he has personal knowledge of the matters set forth in the affidavit.

THE COURT: I do not think that I could possibly accept that as an assertion that he has personal knowledge of everything set forth in this affidavit, because we've gone through a number of places where it's clear that he could not have. And so if he has made a general statement like that over in paragraph 1 and I can see from context it could not possibly be that he has personal knowledge of everything in another statement, then I don't read that general assertion in paragraph 1 as sufficient to apply to every other statement made in this document. That's just not an appropriate way to interpret the evidence.

Now, here he says: "Numerous other events during this period confirmed our belief." He doesn't identify for me what the events are, so I cannot tell whether he has personal knowledge, and I do not read this affidavit because **[3-118]** of the statement over in paragraph 1 as saying that he had personal knowledge of every other event he is referring to in paragraph 28.

MR. DETKIN: Excuse me, your Honor. I believe the rest of paragraph 28 sets forth the events referred to in that opening paragraph, that this event—I'm sorry—opening sentence—that is meant by way of an introductory sentence to the rest of the paragraph.

THE COURT: All right. In other words, you're asking me to read the introductory sentence as not referring to any other event than the ones that are referred to later in the paragraph.

MR. DETKIN: That's correct, your Honor.

MR. GUTMAN: If I might expand on the objection here a little bit, your Honor, he refers to declarations of other people that were filed in this case as if he's adopting what they have said. There's no evidence here that he knew it at the time.

THE COURT: All right. Now, let's take sentence 2. Are you telling me I should read this affidavit as saying that he was present in every technical discussion that occurred between the two companies?

MR. DETKIN: No, but you can read this as saying that he as an officer of the company knew that there were technical discussions between the two companies. They **[3-119]** are set forth more fully in the declarations referred to.

THE COURT: Well, that doesn't give me—what declarations specifically?

MR. DETKIN: There's a parenthetical saying: "See Declarations of Richard Schwartz and Paul Gross." Those are also contained within this submission to your Honor and have already been moved into evidence.

MR. GUTMAN: The problem with that reference, your Honor, is that these were declarations that were filed in this case on the summary judgment motion.

THE COURT: Are they before me in this trial?

MR. DETKIN: Yes, they are, your Honor.

MR. GUTMAN: But my point, your Honor, is, as of the time he wrote this declaration, he may well have seen their declarations and therefore at least at that time had been aware of the contents of their declaration even though as to him it still would be hearsay, but this doesn't say that he knew it as of the time when it would have mattered, namely at the time it was going on, and if he didn't, then it's not relevant. And I certainly would not want the Court to think that this was saying that he knew it at the time when in fact it doesn't say that and there's no support for that.

And indeed, the reference to Release 3.1, the last sentence in that paragraph, "Finally, Lotus' own **[3-120]** advertising, a sample of which is attached as Exhibit H, identifies 1-2-3-compatible menus as a key feature of spreadsheet software in

comparing 1-2-3, Quattro Pro, and other leading spreadsheets," as your Honor will see if you look at the chart, that's the chart about Release 3.1 of Lotus 1-2-3 which didn't come out till the fall of 1990, well after the events and whatever Mr. Kohn talked about. So if this is just an after-the-fact summary judgment vintage collection of hearsay that he is bundling for the Court as opposed to a statement of what he knew in the relevant time period which—and it does appear to be that—then I don't think it's relevant and we certainly ought to not have it admitted for the wrong reason.

MR. DETKIN: Your Honor, Mr. Reback argued eloquently this morning regarding Lotus' entire course of conduct throughout the relevant period as to their continuing course or pattern of deception in terms of misleading Borland, so even though 3.1 may have come out later—I'll be honest, I don't know the exact date—then that would still fall in—that would still be relevant to that course of conduct.

THE COURT: But it wouldn't be relevant to his state of mind as of an earlier point in time. That's what I understand the objection to be here.

MR. GUTMAN: Yes, your Honor. And the first **[3-121]** sentence says: "Numerous other events during this period," and if in fact this was information—

THE COURT: Do you agree that the "this period" that's referred to in sentence 1 is the period that did not incorporate the time when 3.1 was released?

MR. DETKIN: Your Honor, we're not sure about that. If I could just have a moment to confer with my client.

THE COURT: All right.

(Discussion off the record.)

MR. DETKIN: Your Honor, I have been convinced that this was essentially poor draftsmanship. The point is still a relevant one. It probably does not belong with this paragraph with the introduction referring to a particular period of time. So I would argue to you that it is still relevant to the course of conduct, but I would agree that does not fall within the relevant period discussed in that introductory sentence.

THE COURT: Well, then I have to strike the last sentence, don't I? Do you still claim that I should receive Exhibit H, and if so, what's the argument for it?

MR. REBACK: May we just confer for a moment, sir?

THE COURT: You may.

(Discussion off the record.)

[3-122] MR. DETKIN: Your Honor, there is another solution, which would be to strike the first sentence which is troublesome in the first place. That would take out the period.

THE COURT: No, I don't—listen, that's changing the witness's testimony. I can't do that.

MR. DETKIN: Your Honor, I can't add anything more in its defense and therefore bow to your ruling.

THE COURT: Well, I think, then, I have to strike the last sentence in Exhibit H. Now, as to the rest, I receive it as bearing upon the witness's state of mind, but not for any other purpose.

MR. DETKIN: That's fine, your Honor.

MR. GUTMAN: Your Honor?

THE COURT: Yes?

MR. GUTMAN: I'm not sure that I understand as it's sorted out that this is being admitted for the witness's frame of mind as to what time period, because we still—

THE COURT: Well, "during this period" refers to the time period before the commencement of this lawsuit, I take it, at the least. Is it more restricted than that?

MR. GUTMAN: That's the problem. We don't really know from reading it. If we look at the paragraph above—

[3-123] THE COURT: Well, I'm not receiving it with respect to events occurring after this lawsuit was commenced. I take it it's not being offered for that purpose.

MR. DETKIN: That's fine, your Honor.

THE COURT: All right.

MR. GUTMAN: Okay. Your Honor, on paragraph 29, the first sentence, it's just argument.

MR. DETKIN: Your Honor, I can see that is argument and it's merely intended as an introduction to the rest of the paragraph.

THE COURT: Well, the first sentence is struck; not evidence. Third sentence?

MR. DETKIN: It probably would be the fourth sentence, your Honor.

MR. GUTMAN: Actually I think it's third and fourth and we've got the royal "we" again as well. The point is that if the provisions are relevant, they should be before the Court rather than being described thirdhand by this witness.

MR. DETKIN: Your Honor, that assumes there is a writing for all these licenses and assuming that the writing is available, neither of which is shown here. This is just simply referring generally to the types of indemnities that Borland uses with third parties. And as **[3-124]** for the royal "we," I can't imagine an instance where it would be more appropriate when you're talking about the testimony of the general counsel who's in charge of intellectual property licensing to testify about what the licensing contains.

THE COURT: I'll overrule the objection to the third sentence.

MR. GUTMAN: And the fourth, your Honor?

THE COURT: And the fourth sentence.

MR. GUTMAN: Paragraph 30. We have objected to the fourth, sixth, seventh, eighth sentences and Exhibit I, and I hope we've counted correctly in terms of picking the right sentences.

MR. DETKIN: Taking them individually, your Honor, the fourth sentence says that Mr. Kohn contacted outside counsel in early 1990 having read conflicting press accounts. That's his action. I don't see what the hearsay would be.

MR. GUTMAN: Well, he's speaking to the content of the press accounts.

MR. DETKIN: If it's the conflicting press accounts that bothers Mr. Gutman, I'll concede that it was Mr. Kohn's belief that the press accounts were conflicting.

MR. GUTMAN: And concerning the discussion of Quattro Pro on a truck. The sentence recites or purports to **[3-125]** recite elements of the content of these out-of-court statements. If they're only offering it for his state of mind again,

your Honor, well, then I guess it's a different matter, but if they are asking the Court to take this as reflecting something about what—

THE COURT: Are you offering this for anything beyond his state of mind?

MR. DETKIN: No, your Honor, that's fine.

THE COURT: All right. I receive it only as to his state of mind.

MR. GUTMAN: And again, in the sentences that follow, your Honor, there are references to the content of the transcript, the content of what was indicated, what the article said, all of which is hearsay unless this is just his state of mind.

MR. DETKIN: Your Honor, this is for state of mind. After all, Mr. Kohn's state of mind is relevant. However, I would point out that the "Computer World" article is reprinted, and I'm not going to say that what's in there is not hearsay, but to the extent that Mr. Kohn is saying this was in the "Computer World" article, there it is for you.

THE COURT: All right. I receive it only as to Kohn's state of mind.

MR. GUTMAN: Including Exhibit I, your Honor?

[3-126] THE COURT: Yes.

MR. GUTMAN: Thank you. Paragraphs 31 and 32 on the grounds stated. It also looks a lot like argument.

THE COURT: I take it that also is being offered as to Kohn's state of mind. Is that right?

MR. DETKIN: Your Honor, it was offered for Borland's state of mind. However, I understand that you've ruled that he can only speak as to himself and that that same ruling would apply here.

THE COURT: Yes.

MR. DETKIN: That would be for paragraph 31. For paragraph 32, it's an affirmative statement that Borland has spent millions of dollars in acquiring technology. Again, he's a corporate officer, corporate secretary. I believe it is appropriate for him to provide this testimony.

MR. GUTMAN: In reliance upon Lotus' conduct?

THE COURT: Well, you see, there are some problems here now. With respect to the amount spent on the acquisition, I'm inclined to think I should receive it. As to his attribution of causation, "development expenses directly attributable," et cetera, and as to his first sentence, "In reliance upon," I think he can speak only for himself. So those parts—and I have the same problem with "marketing expenses directly attributable." Without the **[3-127]** basis for that conclusion, I can accept it as his state of mind, but to go beyond that, I think I don't have an adequate basis laid here for me to find that he knows or that I have a reasoned basis for understanding what his standard of direct attribution is. When a witness is speaking especially on a causation issue and doesn't tell me what standard of causation he is applying, I don't understand what it means.

MR. DETKIN: I understand your ruling.

THE COURT: So with those limitations, I think I can only receive that part of it as to his state of mind.

MR. GUTMAN: As to paragraph 33, your Honor, in rereading it I'm not sure that these objections only go to the first and fourth sentences. I think they probably go to the entire paragraph in that it's speculation, improper opinion, hearsay.

MR. DETKIN: Your Honor, these are all—this is the statement of the general counsel of Borland stating what he would have done had certain events occurred. Clearly as the general counsel and vice-president he is the person who would have taken these steps and I believe it's perfectly appropriate for this gentleman to be providing this testimony.

MR. GUTMAN: There's no evidence that he had the authority to make these decisions and he's speaking **[3-128]** hypothetically and specutively.

THE COURT: I don't think I can take it beyond his own state of mind. I receive it for that purpose only.

MR. GUTMAN: Thank you, your Honor.

THE COURT: Now, does that complete Kohn's?

MR. DETKIN: Yes, that completes Mr. Kohn's first affidavit. There are four other documents which are part of the equitable defenses submission.

THE COURT: That's the next page, the Exhibits 24, 34, 36 and 37?

MR. DETKIN: That's correct, your Honor. I believe we can deal with these—these fall into two categories.

* * *

[3-129] MR. DETKIN: Your Honor, as for Exhibits 34, 36 and 37, these are documents that—two of them are speeches given by Mr. Manzi and one is a memo that was circulated by Mr. Manzi. Mr. Manzi, of course, is the CEO of Lotus.

Now, we submitted all this last Friday based on the parties' representations that they were standing on the record on summary judgment. I received their objections to **[3-130]** these documents this morning just prior to the hearing. There's clearly foundation in Mr. Manzi's deposition for these documents. He admits that he gave these speeches and circulated the memo. In fact, I find it ironic they're objecting to Exhibit 34. It's a memo from Mr. Manzi that Mr. Gutman on the record stated in the Paperback deposition was exactly that, a memo written by Mr. Manzi.

MR. GUTMAN: If I may, your Honor, our objection to this—if I can save my friend some trouble, our objection to this is not on grounds of foundation. There is no quarrel that this was a memorandum written by Mr. Manzi.

The question, principal question—I suppose there is hearsay content to it; that is, another layer of hearsay that the fact that it was written by our CEO does not cure. But the other problem and probably the bigger problem is one of relevance, because if they're offering this for their affirmative defense case, I can't conceive of what relevance to that case it could have since by definition this is a document they never saw. The legend on the bottom says: "Lotus - Strictly Private," which as the deposition testimony reveals is the highest level of confidential treatment within the company. It bears also the designation "Specially Confidential" pursuant to the confidentiality stipulation in *Paperback*, and that **[3-131]** designation was carried over into this case. Indeed, the last lines of the text, if I may read them without waiving

confidential treatment of anything else in the document, say: "It is extremely important that we do not comment on the suits outside the company. If you are asked about them, please make sure you do not offer your comments, but only say that you cannot comment because the matter is still in litigation."

That being the case, I don't understand how this document, which absent a burglary or something I can't imagine they are going to claim and there's certainly nothing in the record to suggest they ever saw, can have any relevance whatsoever to any of their affirmative defenses.

MR. DETKIN: Well, your Honor, Mr. Gutman is arguing relevance. We offer it for a limited purpose which is in the first sentence of text. It states that Lotus filed suit against two, quote, clone, close quote manufacturers, and that goes to our defense as to—actually, I apologize.

Can I refer to this in court?

MR. GUTMAN: No. Your Honor, precisely the reason that out of all of our relevance objections this is one of a handful we have selected to argue live to your Honor is, if we are right about relevance, then we ought not to have this specially confidential memorandum from the CEO **[3-132]** of the company part of the file. We understand how your Honor likes to treat confidential materials and that you don't like them lying around, and in this instance, if we are right about our objection, the easiest solution is, your Honor can certainly look at it for purposes of ruling and all of that, but we can then take it away and we have one less document to worry about.

THE COURT: All right. Now, if I understand your statement of the purpose for which you're offering it, I think what I should do is to say to you that a statement at this time that the products of Paperback and Mosaic were clones is relevant to your claims about earlier periods of time. I take it for that limited purpose, but that means I don't need the rest of the document in the record and so I should simply state on the record that I receive that limited part for this purpose and the rest of it may be withdrawn from the record so the confidentiality is preserved.

MR. GUTMAN: Thank you, your Honor.

MR. DETKIN: That's fine, your Honor.

THE COURT: All right.

MR. DETKIN: Your Honor, Exhibit 36 is the speech from the Royal Sonesta Hotel that was the subject of one filing before your Honor. This is the outlined version. It is cited in our summary judgment papers and its **[3-133]** relevance is set forth there and discussed in detail. Again, I'm handcuffed from saying what contents of it we consider relevant in open court.

MR. GUTMAN: Without getting into the content, your Honor, and again, since these are very confidential materials, an internal speech, this was not published, this was Mr. Manzi speaking to his people, our—

MR. REBACK: There's only one place that we care about, your Honor—

MR. GUTMAN: If I may finish my point, and I would ask my brother not to read into the record in open court—

MR. REBACK: I'm not going to read anything. I'm just going to tell the judge what we're referring to.

MR. GUTMAN: The point—this speech was made five months, six months after we had sued them, your Honor. That being the case, we have a hard time understanding how it could even arguably be relevant regardless of its content to any of their affirmative defenses, because it came later.

MR. DETKIN: Your Honor, the relevant portion confirms Lotus' view—

MR. REBACK: Just a second. Your Honor, the part we would call your direction to is on the page numbered at the bottom 049168, and if you go about a third of the way up that page at the line starting "Competition," that's the **[3-134]** point to which we're referring, sir.

MR. GUTMAN: We've still not heard any theory of relevance.

MR. DETKIN: Read our summary judgment papers.

THE COURT: Give me just a moment.

MR. REBACK: Yes, sir.

(Pause - Court is reading.)

THE COURT: From that line how far, how many lines?

MR. REBACK: To the bottom of that page, sir.

(Pause - Court is reading.)

THE COURT: All right. Now, what is the objection?

MR. GUTMAN: The objection, your Honor, is that that statement made six months after the lawsuit was filed internally at Lotus with there being no evidence in the record, unless somehow it happened since the litigation started in this process, no evidence that anyone at Borland ever saw that or heard it or was familiar with its content, our position is that that can't be relevant to any of their affirmative defenses.

THE COURT: Well, why isn't it an admission as to state of mind and objectives and conduct at an earlier period of time which is relevant?

MR. GUTMAN: I'm not sure that—with waiver **[3-135]** out of the case, your Honor, I don't mean to quarrel with the Court, but it's not clear to me that on estoppel or laches our state of mind would be relevant.

THE COURT: Well, I overrule the objection and I will receive in evidence the portion of page 049168 beginning with the words "Competition from Borland" and extending to the bottom of the page.

* * *

[3-163] MR. GUTMAN: Leaving that aside, I believe the only thing we have left is our objections to the Declaration of Eugene Buechele, which I am told is pronounced Buechele.

THE COURT: And under what tab?

MR. DETKIN: Which declaration is that, Hank, first or third?

MR. GUTMAN: Okay. Well, as I understand, there were three originally, the second was just dropped by Borland, so I'm working on the first.

THE COURT: And under what tab?

MR. GUTMAN: It's tab 1 of Volume 3B in your Honor's set.

[3-164] And the first objection, your Honor, is paragraph 4, second sentence. As stated on page 4 of our objections, it's lack of personal knowledge and an improper opinion without adequate basis. I would also add, your Honor, in light of the

fact of our stipulation concerning long prompts, that I don't know what the relevance of this is, either.

MR. DETKIN: Well, we're going to go back to relevance.

MR. GUTMAN: No, I raise that because as I understood this, the purpose of this particular Buechele declaration was to try and establish that VisiPlot and VisiTrend had long prompts. If that was what this was about, that issue I think is gone because we've stipulated that long prompts are no longer in the case and are not material to any issue that is in the case, and if I'm right in that and my friends from Borland agree with that, they might choose to withdraw this declaration and we can save your Honor—

MR. REBACK: No, sir.

MR. GUTMAN: Then I have my objection. And I would add relevance—

THE COURT: Well, what does the witness mean by the control mechanism of the 1-2-3 user interface? I understand the two-line moving cursor, but if he's talking **[3-165]** about something broader than that, then I don't know what he's talking about and I need to understand that in order to evaluate both the evidence and this objection.

MR. REBACK: Yes, sir. The definition of control mechanism is provided in the summary judgment brief, however, and that's the problem.

THE COURT: Wait a minute. How do I know that's the definition he is using?

MR. REBACK: Well, let me say that for the purposes of what he is using, if you go down to paragraph 5 you see the control mechanism similarities and they are listed there. The one we care about, your Honor, is the fifth one down, the ordering of commands by anticipated frequency of use. Forget what he says about Lotus. What we care about is that "Yes, under VisiTrend/Plot." That's what we care about.

MR. GUTMAN: Your Honor, now that I understand the purpose of the offer, there is the problem not only that the witness never says what he means by control mechanism and Mr. Reback cannot fill in that definition for him, but where's the basis—where is the showing of firsthand knowledge that the

commands in both of these products were ordered based on anticipated frequency of use?

MR. REBACK: What's being proferred is the order of commands in VisiTrend/Plot. If you look paragraph **[3-166]** 2, the last sentence, he says that he was in charge of that product and in charge of the development of that product.

MR. GUTMAN: In paragraph 3 he says it was designed by Mitch Kapor.

MR. REBACK: Yes, and licensed to VisiCorp, and he says that he was the person in charge of that. When Mitch Kapor left, he was in charge of the ongoing development of VisiTrend/Plot.

THE COURT: All right. I will overrule the objection. I take account of the arguments you're making, Mr. Gutman, in my weighing and evaluating the evidence, but I overrule the objection.

MR. GUTMAN: Thank you, your Honor.

Paragraphs—oh, I'm sorry. Was this—we moved from the second sentence—we moved, your Honor, from the second sentence of paragraph 4 into the fifth item on their chart. I had objections to both. Was your Honor ruling on both?

THE COURT: Well, I hadn't, but I will since it's been explained to me that the two were meant to go with each other.

MR. GUTMAN: Am I correct in understanding that the offer in the second sentence is now limited to that fourth point and then only VisiPlot/VisiTrend rather than 1-2-3? Because otherwise—

[3-167] THE COURT: Is that correct?

MR. GUTMAN: —we have the control mechanism problem.

MR. REBACK: For these purposes, yes, that's right, for this phase of trial, your Honor.

MR. DETKIN: Just to correct one thing Mr. Gutman said. I believe he said the fourth point. I think he meant the fifth point.

MR. GUTMAN: Fifth point, that's right. Thank you .

The next objection, your Honor, is paragraph 6.

MR. REBACK: I don't understand the objection. He's describing the product that he's in charge of developing.

MR. GUTMAN: He was at the licensee, number one. Number two, he is expressing an opinion concerning similarity. Number three, he has not shown the qualifications to express that opinion. Number four, he has not shown the sufficient familiarity—even if we assume for these purposes that he knew VisiPlot/VisiTrend, because he works at the licensee rather than at the developer, he has still not shown sufficient familiarity with 1-2-3 to be expressing the opinion he does concerning similarity.

THE COURT: Well, all right. Now, let me make clear to you that I am not receiving this as an opinion on **[3-168]** any legal issue, so when he uses "similar," it is not similar in any legal sense. This is a factual sense, lay sense of similar. I overrule the objection on the understanding that that's the way I am reading "similar."

MR. GUTMAN: Thank you, your Honor.

Your Honor, the next point's the third Buechele declaration.

THE COURT: Is that tab 3?

MR. GUTMAN: Tab 3, your Honor.

THE COURT: All right.

MR. DETKIN: Your Honor, this went to long prompts and I would suggest that the objection is moot.

MR. GUTMAN: Well, if it went to long prompts, I would suggest the evidence is moot.

MR. DETKIN: Fine. You withdraw your evidence on copying?

THE COURT: Well, wait a minute. Are you withdrawing this proffer since it went to long prompts?

MR. REBACK: May I just take a moment to look at it, sir?

(Pause.)

MR. REBACK: The reason I can't withdraw the proffer is because I believe—excuse me just a second, your Honor.

(Defense counsel confer.)

[3-169] MR. REBACK: The reason I can't withdraw the proffer is that Mr. Gutman put into evidence VP-Planner Plus

on the issue of long prompts, and if he'll take that out, I'll take this out.

MR. GUTMAN: Your Honor, I believe what Mr. Reback is referring to was either a declaration or a deposition of Jim Stephenson from the *Paperback* case which did address the long prompt point and which we removed, that evidence we deleted, so if that's what he's talking about, it's already out.

MR. REBACK: No, the product you put in.

MR. GUTMAN: If the product is in, your Honor, it's in as another—excuse me a moment. I'm not sure if VP-Planner Plus is in.

(Plaintiff's counsel confer.)

MR. GUTMAN: Your Honor, VP-Planner Plus is already out. Its only offer was to support a chart we had prepared. If we'd had to litigate merger of long prompts, we had a chart that was going to show it.

MR. REBACK: Based on that proffer, your Honor, I withdraw this.

* * *

[3-181]

MR. REBACK: No. I'm glad your Honor raised that. I don't want the Key Reader portion closed, but I don't think we should be going back for designations and counter-designations on the other things.

THE COURT: No, I understand that, but what I mean is, I can't decide the matter —

MR. REBACK: Yes, sir, I understand that. Your Honor said that you think that's the best procedure and we agree. That's fine.

THE COURT: Okay. That's very good.

MR. REBACK: I do want to make it clear we're not going to have any more designations, for example, on equitable defenses and things like that.

Okay. The only other thing I want to do is, I want to make sure—we moved this into evidence, there was no precise objection on it, and I don't want there to be any misunderstanding. This is a Wall Street Journal advertisement from

Lotus bearing the Lotus copyright notice. It's found in Volume IV of your book, your Honor, at tab B-5, and what I'd like to do is hand the true copy of it up because you have a smaller copy which is harder to read.

THE COURT: Now, was there objection to this?

MR. GUTMAN: Yes, your Honor. We understood it to be a Key Reader document, but if I understand now that **[3-182]** Mr. Reback is offering it in support of fair use on the non-Key Reader case, I have an objection of relevance, because we don't believe that issue is in this case because they never raised it in a timely fashion in their answer.

THE COURT: All right. Now, I have ruled on a similar objection before by saying I overrule that objection. I am not making any decision at this point on the relevance issue or on the substantive arguments between you as to whether the fair use in the sense that Mr. Reback is using it is before me in this case as distinguished from the more limited question about fair use as you understood my question to you in the January conference.

MR. GUTMAN: No, I understand, your Honor, and if I may just make one further clarification responding to something that Mr. Reback said.

Since it was our position—it was our understanding without question prior to Mr. Reback's argument today and it is still our position that fair use has not been raised as a defense on anything other than the supplemental complaint, we have not put in any case on fair use.

Now, I don't believe that what he has offered, if I've understood him correctly, on fair use apart from the supplemental complaint would make out a fair use defense, so I think not only has he failed to plead it in a timely **[3-183]** fashion and therefore waived, but I think if he is saying that he has now offered everything he wants to offer on fair use and he's rested on it, I would also argue that we would be entitled to a directed verdict. But I just want the record to be clear so that there's no —

THE COURT: I understand that and that argument you may—it has been made to me. I consider it.

MR. GUTMAN: And if your Honor does decide to let him argue fair use in this case, we would like the opportunity to put on proof if that's necessary. We are not resting on that issue. That's why I wanted to make it clear now.

THE COURT: Well, now, maybe we better pursue that a moment longer.

Mr. Reback, how do you respond to the proposition that if I do not rule that by the directed verdict standard that what you have offered on fair use as you have defined it today in your argument, then Lotus has not had the opportunity to present its evidence on this subject?

MR. REBACK: Well, I had thought their evidence was in, your Honor, because they put in the pages—the agreements on the fair use point your Honor raised. They put a bunch of evidence in on copying.

THE COURT: Well, on that proposition I did not in my questions—even though you've interpreted **[3-184]** otherwise in your argument to me today, I did not in my questions in the January conference raise with you any question about fair use in the sense in which you have used it. I was talking about the question whether the fair use by a third-party would affect the issues in this case if Borland instead of copying directly copied from something that was a fair use by a third-party. Now, I thought I made that very clear in the questions I was asking.

MR. REBACK: You did ask questions directed to that, your Honor.

THE COURT: Yes. So I never asked a question or suggested to you that there was any fair use issue in this case in the sense in which you have developed it in your argument today. So it does seem to me that if you want to contend that fair use in the sense you have argued to me today is in this case, then I have to give Lotus an opportunity. I certainly didn't give them notice that anything like that was going on in this trial. I have to give them an opportunity to offer their evidence on that subject.

MR. REBACK: Well, if that is your Honor's desire and that's what you think —

THE COURT: Well, it's not my desire. It's the only way I think we can have a fair trial.

MR. REBACK: The reason that we only put a **[3-185]** small portion of this in our brief is because—I guess it's the point that Mr. Gutman was making. If your Honor is not interested in hearing this, doesn't believe it's meritorious or is not ready to make a decision on it —

THE COURT: No, you're putting it entirely wrong. It's not my desire. I make rulings about what I think is appropriate for a fair trial in the case. My desire is totally irrelevant. What I would wish is not the basis of my rulings. I make rulings that I believe are correct according to the law and the facts in the submissions before me.

Now, I have said to you that in order to have a fair trial on the fair use issue in the sense in which you have raised it in your argument today, I can only do that it seems to me when I have given notice to the parties that that's what we're doing and have heard all the evidence that both sides want to offer on that subject.

MR. REBACK: All right. Then your Honor will permit Lotus to make the same kind of argument that I make today, to designate the portions of the record that they would like to use for that purpose. I understand your Honor's ruling. That's fine.

THE COURT: I'll tell you what I think would be a more efficient way of proceeding. That is again going to delay my reaching the other issues that I would otherwise **[3-186]** decide.

MR. REBACK: Yes, sir.

THE COURT: I think it probably makes more sense to say that issue is not on trial in this Phase I, but if you want it to be on trial in this Phase I, then I don't think I should be deciding anything until we have reopened the record to get all of the evidence that both sides want to offer on fair use in that sense or else I have heard arguments and ruled that it comes too late, which I take it is another position Lotus is asserting.

MR. REBACK: Well —

THE COURT: You know, I haven't even heard your argument as to why I should allow you to raise this issue this late in the proceedings. You suggested to me this morning that I raised it. I don't think that's a fair reading of what I asked in the January conference. I didn't raise it, the issue that you're arguing. I raised a very different kind and much narrower question.

MR. REBACK: I understand, your Honor. I guess we were prompted by your Honor's comments at the time that you wanted to resolve everything on the issue and I guess we just took a broader view than your Honor had intended. But what I would like to do, I think, is to get that matter resolved as well, because your Honor has indicated that he only wants to hear all of this once.

[3-187] THE COURT: Well, then I think probably the first step I should take is to receive your written briefs very promptly on whether it's appropriate for me to take that issue in this case when it's first raised at this late stage.

MR. REBACK: When would you like us to submit the briefs, sir?

THE COURT: As quickly as you can.

MR. REBACK: Two weeks?

THE COURT: Simultaneous submissions —

MR. REBACK: Sure, with a week to reply.

THE COURT: All right. Simultaneous submissions in two weeks and a week to reply.

MR. GUTMAN: If I could just check the calendar, your Honor, because I'm going to be away on vacation next week and I want to make sure that I'm not —

MR. REBACK: Mr. Gutman, I'm going to be on vacation too. What's your proposal, Mr. Gutman?

MR. GUTMAN: If I could consult the calendar, I do want to make sure that I have some time back before the brief is due. And under the stipulation we already signed, your Honor, our submission on Key Reader is due the 19th, so we do have another assignment during that period of time.

THE COURT: Well, the two weeks would have been the 18th. I would suggest we do them both on the 19th **[3-188]** instead of one the 18th and one the 19th.

MR. GUTMAN: If I could just consult with my teammates to be sure I'm not going to be burning out anybody here.

MR. REBACK: Actually I think we both have a problem. Mr. Gutman says he has vacation for the week of the 8th, and Mr. Detkin and I had assumed we'd be finished with this. We have vacation the week of the 15th. Perhaps the first submission could be made the 26th and a reply the week later.

THE COURT: Twenty-sixth of February and the 5th of March.

MR. REBACK: Yes, sir, please.

MR. GUTMAN: That's fine, your Honor.

THE COURT: All right.

MR. GUTMAN: I was going to suggest—now, as I understand it, this brief is addressed to the question of whether it's too late.

There's another possibility that had occurred to me, and that was, if Mr. Reback has made his showing; that is, if he rested or is about to utter the words "I rest" today on this defense on the assumption that it's in the case, I could at the same time in the alternative make a motion for a directed verdict on that. This is an issue on which—

[3-189] THE COURT: Yes, or to be more precise, a Rule 50(b) motion I think it is now with the amendment of the rules. I hope I'm remembering correctly.

MR. GUTMAN: That's what I was referring to, your Honor.

MR. REBACK: The problem with that is it may not resolve everything.

THE COURT: I beg you pardon?

MR. REBACK: It may not resolve everything.

THE COURT: Well, that's true. If I don't sustain it, it does not. We go on from there.

MR. REBACK: Let's just go on the schedule we discussed.

MR. GUTMAN: But as I understand it, the schedule we discussed would only be resolving the question of whether it's too late for him to raise the defense. What I was proposing to

do was to resolve two issues in the same round of briefs; (a) is it too late. If the answer to that is yes, it's too late, then your Honor never has to reach the second question. B, if it's not too late, is it in any event too little, which your Honor could rule on, and the only way —

THE COURT: Mr. Reback, that makes sense to me unless you're saying that if I'm going to try the issue, I'm going to let you have it this late, you want to put **[3-190]** something more in.

MR. REBACK: No, that suggestion I think makes sense to me as well, your Honor, but if Mr. Gutman is intending to file something in support of his directed verdict, I would ask to be able to file something in response.

THE COURT: Well, of course. What I was proposing—I think what Mr. Gutman was proposing is, we've got the same schedule. He would include that in his filing on the 26th of February. You would include it in your response on the 5th of March.

MR. REBACK: Okay. If I have to do both of those things, can I get a two-week reply on that one, your Honor, taking me to March 12th? Because Mr. Gutman will have had three or four weeks to —

THE COURT: Well, do you want to just make all of the responsive submissions, both of you, you file your original submissions on the 26th and responsive submissions on March 12th?

MR. REBACK: That's fine, your Honor.

THE COURT: All right, that's fine.

MR. GUTMAN: Thank you, your Honor.

MR. REBACK: I don't want to leave without moving this—making sure this is in evidence.

THE COURT: I have received that in **[3-191]** evidence—

MR. REBACK: Thank you, sir.

THE COURT: — for the limited purpose, yes.

MR. GUTMAN: That's fine, your Honor.

THE COURT: All right.

MR. GUTMAN: And if I'm moving for a directed verdict, I would have wanted it.

THE COURT All right. Now, am I correct then in understanding that except for the Key Reader evidence that we have yet to resolve, both sides are resting at this point?

MR. GUTMAN: In terms of the evidence, your Honor, yes. There is one further matter I wanted to raise, but it is not an evidentiary submission.

THE COURT: All right.

MR. REBACK: Can I hear that, please?

THE COURT: Both sides—well, all right. Do you want to hear that first? Okay. What is it?

MR. GUTMAN: Your Honor, it is a small matter and it will take only a couple of minutes, but there is a piece of argument that I would have delivered when we did our closing arguments but for the fact that I couldn't when we delivered our closing arguments because it turns on rulings your Honor made after the argument with respect to Mr. Kohn's declaration. And what I would ask is for the **[3-192]** Court's patience after Mr. Reback rests and all of that is clear to supplement my closing argument simply to point out something which I think your Honor should have in mind in light of the rulings that were made on the Kohn declaration which I obviously couldn't argue this morning because your Honor hadn't yet made those rulings. I just want to bring something to —

THE COURT: Well, of course if I do that, I'll have to give Mr. Reback an opportunity for rebuttal on that.

MR. GUTMAN: I understand that, your Honor. It should take a couple of minutes.

THE COURT: All right. Any objection to that, Mr. Reback?

MR. REBACK: Well, I do object on the grounds that I went, he went, and then I didn't take a reply, but —

THE COURT: Well, I think, though, it is true that I accommodated you by allowing this to occur before the record was clear and that's what this request relates to, so I think should allow you both to speak again in that respect.

MR. REBACK: Very well, sir.

MR. GUTMAN: Thank you, your Honor.

THE COURT: All right. Now, I'm about to hear that argument. Is it true that both parties now rest except as to the

additional evidence on Key Reader which I am to **[3-193]** receive under the agreement?

MR. REBACK: Just to be clear, this document, the Wall Street Journal, is in the record on fair use?

THE COURT: Well, Mr. Gutman, do you have any problem with that, that the Wall Street Journal item is in on the fair use evidence? This is not a question of its sufficiency.

MR. GUTMAN: No, I understand. Your Honor, for purposes of my motion for a Rule 50 motion on fair use, certainly it may be considered without waiving any other relevance objections that there may be to it.

THE COURT: Well, as to relevance objections, of course, I take that along with the case.

MR. GUTMAN: This is authentic document. We have no objections other than relevance.

THE COURT: Okay. It's in.

MR. REBACK: Thank you, sir.

THE COURT: All right.

MR. REBACK: The only—I hate to do this. The only question I had is, everything would be closed with the exception of Key Reader? That's what your Honor is asking us?

THE COURT: Yes.

MR. REBACK: Okay. Now, in the event that Mr. Gutman's motion is denied for directed verdict on the fair **[3-194]** use, what then?

THE COURT: If both his motion that it's untimely and his motion for judgment at the close of the counterclaimant's evidence is denied, then he will have an opportunity to present evidence on the Key Reader before I make a decision on that question.

MR. REBACK: Okay. I was just asking about the status of evidence on fair use. That was my question.

THE COURT: Oh, wait a minute.

MR. GUTMAN: I think your Honor meant to say fair use.

THE COURT: I meant to say fair use when I said Key Reader.

MR. REBACK: Okay. Well, what I'm asking is, am I then foreclosed from submitting rebuttal evidence? Because I don't

think I should be foreclosed if he's going to submit some more evidence.

THE COURT: Do you want to respond to that, Mr. Gutman?

MR. GUTMAN: Well, your Honor, I think we would be in the position, if I understand it correctly—he has put on a trial of his affirmative case because he has the burden on this affirmative defense on fair use. I am moving at the close of that. If I lose that motion, then I would put on my case and I would think that his right, if **[3-195]** any, to put on anything after that would be the way it would work in any other trial to the extent—

THE COURT: Which is to rebut your evidence as long as it's not repetitive of what he put on.

MR. GUTMAN: Exactly. As long as it fits in the scope of proper rebuttal he would have it, but nothing else.

THE COURT: All right. I think that's appropriate.

MR. REBACK: I think so too, sir.

THE COURT: All right, fine. All right. I'11 hear your supplemental argument. Well, you rest also, I take it.

MR. GUTMAN: Yes, your Honor.

THE COURT: All right, fine.

MR. GUTMAN: Your Honor, I'll try to do this quickly and I'm sorry it doesn't quite flow the way I would have hoped it would have if it had fit in integrally with the rest.

But your Honor will recall that in going through the Kohn declaration today, a great many of the paragraphs, including a number that seemed to have much substance, ended up being stripped or reduced in the offer to being offered only for Mr. Kohn's state of mind. The reason that's significant is laid out in some deposition testimony of Mr.

* * * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

Courtroom 11
Wednesday, March 31, 1993
Boston, Massachusetts

LOTUS DEVELOPMENT CORPORATION

VS.

BORLAND INTERNATIONAL, INC.

NON-JURY TRIAL PROCEEDINGS
DAY 4

Before:

THE HONORABLE ROBERT E. KEETON
United States District Judge

APPEARANCES:

O'SULLIVAN, GRAEV & KARABELL, by Henry B. Gutman, Esquire, Kerry L. Konrad, Esquire, Joshua H. Epstein, Esquire, and Paul M. O'Connor III, Esquire, 30 Rockefeller Plaza, New York, New York 10112, on behalf of the plaintiff.

WILSON, SONSINI, GOODRICH & ROSATI, by Gary L. Reback, Esquire, Peter N. Detkin, Esquire, Andrew G. Konstantaras, Esquire, and Isabella E. Fu, Esquire, Two Palo Alto Square, Palo Alto, California 94306

—and—

DONNELLY, CONROY & GELHAAR, by Peter E. Gelhaar, Esquire, 176 Federal Street, Boston, Massachusetts 02110, on behalf of the defendant.

TIMOTHY J. WILLETTE, RPR-CM
Official Court Reporter
U.S. District Court—P.O. Box 347
Boston, Massachusetts 02101-0347
(617) 248-0604

[4-5]

MR. GUTMAN: I have no problem with turning them off, your Honor.

THE COURT: All right.

MR. GUTMAN: I'd just as soon leave the computer on, if that doesn't distract anybody.

THE COURT: Well, you could. All right, sure.

MR. GUTMAN: I can turn the Court's monitor off, too, if your Honor would like.

THE COURT: All right.

(Monitors turned off.)

MR. GUTMAN: Your Honor, the motion that we made orally at the close of the Phase I trial in February when Borland's counsel in closing argument raised the fair use defense as to that original complaint which had not previously been pleaded, and indeed their leave to amend was not even sought until a week after that trial, I called it Rule 50. Your Honor correctly corrected us, and I apologize for having called it by the old rule number. It is a 52 (c) motion and we believe that we fall within the rule.

"If during a trial without a jury a party has been fully heard with respect to an issue, and the court finds against the party on that issue, the court may enter judgment as a matter of law against that party on any **[4-6]** claim," et cetera, "that cannot under the controlling law be maintained without a favorable finding on that issue."

Now, the issue in question is fair use on the original complaint where Borland, notwithstanding the fact that the issue wasn't even in the case at the time and subject to our relevance objections, put in their complete evidence, no qualms about it, no reservations. They said, "This is it. We rest." So I think the motion, if I understand the rule correctly, is ripe, because they have put in everything they have on that fair use issue. And it's our position that this is an affirmative defense, it is their burden, and they have failed to meet it.

Now, your Honor's decision of yesterday, memorandum of yesterday, asked us to come in prepared to address the question of what standard applies under the federal rules, and that is a very interesting question. Between yesterday and today, looking at the cases quickly, we weren't able to find a lot of guidance on the point, but we did find some which we believe supports what was our logical assumption, and that is that the standard that applies on a 52(c) motion where there's no jury right implicated, there's no Seventh Amendment question, is not the old directed verdict standard of whether or not a reasonable jury could find to the contrary, but your Honor as sitting as finder of fact. Your Honor has heard all the **[4-7]** evidence from one side, the side with the burden. If your Honor having heard all of that evidence believes that they have failed to make out the defense, that's enough. It's a preponderance of the evidence standard. It is not—it is not beyond what a reasonable jury could find—you know, whether a reasonable jury could not find otherwise. It is not that there are undisputed facts as in summary judgment. I believe logically under the rule—and there is some, not a lot, but there's a little case support for it—that the standard is precisely that. It's a preponderance of the evidence standard.

So if your Honor evaluates everything they've put in on fair use on the original complaint, having rested, and your Honor is not persuaded that they have made out the defense, I believe that a 52(c) provides the Court with a vehicle for dealing with that issue.

Without putting us to the expense and without putting the Court to the waste of time of requiring Lotus to put in its case on fair use to rebut, which as I believe your Honor recognized when we were last here we have not done, and when one considers the timing of their raising of the defense, that isn't at all surprising.

Now, I would now move—if your Honor would like—I mean, we can talk more about the rule. I was prepared to move to the merits—

[4-8] THE COURT: All right.

MR. GUTMAN: —the merits of their fair use defense.

As I said, the burden is on them. It's an affirmative defense. I don't think there's any question under the case law that it's their burden to prove fair use, and they haven't done it here. Let's begin by looking at the four fair use factors. First, what's the purpose and character of the use.

I don't think there is any dispute on this record that the purpose is commercial, and it's not just vaguely commercial, modestly commercial, but this is the commercial activity of a direct and heated competitor. There's no argument that their purpose in copying the 1-2-3 menu, menu structure, macro language and keystroke sequences was scholarship or research. There's no argument that as in *Sega* it was simply for purposes of internal use or in *Coates-Freeman* where it was internal use, and in *Sega* where they persuaded the court that this was in fact the only way to understand, to learn the unprotected ideas in the program. And this was not for non-commercial home use which was what was at stake in *Galoob*, and that's the other case that they base this on. They place their heaviest reliance on *Sega* and *Galoob*. Neither helps.

In *Galoob*, the issue—the fair use issue was **[4-9]** whether this was a fair use for the people who were using the game

genie in their own homes for non-commercial purposes to speed up the play of the Nintendo game whereby so doing infringing Nintendo's copyrights in those games. And the only claim against *Galoob* that was articulated contributory infringement claim, and the court correctly ruled you can't contributorily infringe on something which is not an infringement by virtue of fair use.

So none of those arguments help them here. It's clearly commercial. And again, unlike *Sega* where the court said the commercial impact if any was minimal, here the commercial-impact is clearly quite meaningful.

The record is full of testimony from both sides concerning the importance of the menus, the importance of macro compatibility. They offered it up as the justification—the purported justification for their behavior. "We had to do it because it was so commercially important." And they were kind enough to even cite testimony from our people, including from Jim Manzi, saying, "Yes, we consider compatibility to be an important issue and we're developing new versions of our own product."

So I don't think there's any question that on the first standard, which is the purpose and character of use, it is commercial, it is competitive, and as the Supreme Court has said repeatedly, that makes it presumptively **[4-10]** unfair. They lose unless they can show no adverse impact, no adverse economic impact on us, which is the fourth factor of the test, but the burden—at this point it becomes almost a double burden—is still on them.

The second factor to consider is the nature of the work. Now, here this is an issue that your Honor has, I think, dealt with in the summary judgment decision to some great extent. What's at issue here are the menus and menu structure of Lotus 1-2-3, keystroke sequences, and the macro language. The Court has already ruled that they are expressive. The Court has already ruled that they are protected with—I'll drop a footnote here for the fact that your Honor did leave open pending trial a question of whether there were limits on that

protection and whether certain elements or subelements might not be, but that this is fundamentally protected I believe was established in the summary judgment decision this past summer.

Now, Borland's argument on the nature of the work based on a misreading of *Sega* is essentially that fair use is a given or there's almost a presumption of fair use if what you're talking about is a computer program because that's a utilitarian work. That's not what *Sega* says. That's not even close to what *Sega*—to what *Sega* says. One may believe that *Sega* is good law or one may believe that *Sega* is bad law, and it still remains to be seen the **[4-11]** extent to which other circuits and the Supreme Court will ultimately feel that *Sega* is good law. But what *Sega* is certainly not is a declaration of open season in fair use on computer software.

The *Sega* court was quite clear to make it—to point out that they were talking about a very precise situation in which the copying was internal, in which the product that was ultimately sold did not contain any of the copied material and in which that copying done for internal purposes was basically akin to research, because under the very specific circumstances there this was the only possible way to learn the unprotected ideas in the work. A very narrow holding, and the court was very careful to make it such. So I don't believe that it stands for the proposition that Borland cites it for, that software is somehow suspect and easy to prove fair use.

The third element is the amount and substantiality of the taking. And at least of the courts—and I believe we've cited these cases in our memo on the subject, your Honor. They equate this factor, this standard, with the quantitative substantiality test, which is part of the copyrightability and part of the infringement test that your Honor has articulated well in this case. There's no question that the taking here was substantial. The menus, menu structure, macro language, keystroke **[4-12]** sequences. Again, I believe your Honor addressed that issue in dealing with the summary judgment motion this summer. This was not a trivial or de minimus taking.

Now, Borland takes your Honor's summary judgment decision this summer and argues that that proves that they win on that factor because your Honor stated that the standard is modest; that is, one need not show a tremendous amount in order to show qualitatively substantial taking. So they take that as saying, "Ah-ha. Well, your Honor held that in fact what was taken here was something very modest and therefore Lotus loses on the third standard." I don't believe that's a fair reading of your Honor's holding this summer and it's certainly not a fair reading of the record here. Again, the same evidence from both sides that they put in about the importance of these aspects of the program, including the importance of macro compatibility, proves that this taking was not insubstantial.

And then the final factor which in the words of the Supreme Court in *Harper & Row* is, quote, undoubtedly the single most important element of fair use.

Now, Borland says that's no longer the law. *Sega* changed all that. I'd be curious to know how the Ninth Circuit gets the ability to do that to the Supreme Court. But in fact if your Honor looks closely at *Sega* and looks closely at *Galoob*, the same language, the same principle, in **[4-13]** the Supreme Court decisions is quoted in each of the Ninth Circuit cases upon which Borland relies. This is the most important factor.

Again, *Harper & Row* goes on to say—and I'm reading at pages 2233 and 2234 of the Supreme Court Reporter edition: "Fair use when properly applied is limited to copying by others which does not materially impair the marketability of the work which is copied."

Same page: "More important, to negate fair use one need only to show that if the challenged use should become widespread, it would adversely affect the potential market for the copyrighted work." So it's not even necessary that the impact be immediate and present. Even the threat of a future potential impact is sufficient under the cases. And in this instance the *Harper & Row* decision is quoting from *Sony*, an earlier Supreme Court decision on the subject involving the

Betamax, which like *Galoob* involved contributory infringement for a home fair use.

All of the cases, your Honor, are to the same effect on this, running through *Sega*, running through *Galoob*, including your Honor's decision in *Coates-Freeman* where your Honor pointed out on page 887 of the Fed. Sup. that because Polaroid was given the chart and could not otherwise have bought a copy, its limited internal use did not even deprive Coates-Freeman of a sale to Polaroid, let **[4-14]** alone to any third party. So in the *Coates-Freeman* case where your Honor found fair use there was not even any evidence or any showing that even a single—so much as a single sale had been diverted or interrupted based on Polaroid's internal use of Coates-Freeman's work.

Now, again, they—they try to distinguish the cases that say that the fourth factor is the most important by saying, "Oh, well, these are artistic works. This is really different. It's a different standard when you're talking about something utilitarian." I would suggest to the Court that whatever one may think of the quality of president—ex-President Ford's writings, his memoirs were not—and indeed some might even argue that in certain aspects they might be fiction, but clearly the court viewed them as a historical work, as a work of non-fiction.

So the notion that this—that the importance, the overwhelming importance, the decisiveness in many cases of the fourth factor is somehow restricted to works of aesthetic art as opposed to works that—that involve non-fiction or involve utilitarian works is a distinction that they have made up and that is nowhere supported in the case law, including the Supreme Court case law.

Now, what have they put in on this fourth factor where they have the burden in order to demonstrate that there's been no adverse impact on us? All they've offered **[4-15]** is a couple of items, a newspaper advertisement, an excerpt from a speech from which they argue that Lotus had a 70 percent share of the spreadsheet market before they started infringing.

Lotus still had a 70 percent share of the spreadsheet market as of this summer, so obviously they've had no effect.

Now, even if one credits those statements that they rely on well beyond what they're worth and we leave aside the questions of whether they're comparing apples and oranges and whether one is talking about units and the other is talking about dollars and whether they are approximate or whether they are even remotely accurate, and even if we leave aside the fact that they don't address the question of, well, what happened in the meantime, were there years during the period when they infringed when they had a very substantial market share taking at our expense, none of which are suggested by the flimsy proof they've put in, the answer is, so what. The question isn't whether our market share is what it was before they started infringing. The question is, have they injured us in any meaningful way. And how can they deny that they have? There's certainly nothing in the record to suggest or to prove that they have not.

They acknowledge that we compete. Do they really contend that none of the sales they've made since 1987 have **[4-16]** been at Lotus' expense, that they're in the Coates-Freeman situation where not a single sale was lost? They have a non-trivial market share. I think if we had to, we could quite easily prove from their mouths claims of nine percent, twelve percent, 24 percent, 30 percent, depending on the time and depending on what suited their purpose at the time.

Now, where did all of that come from? Is it really the case that none of that market share came from us, none of the sales came from us? That when their salesmen and ours every day go head to head in corporate accounts saying, "Buy my product, don't buy theirs," that they've not won a single one of those fights? If that's the case, they have not proven it, and they have to, and they've failed. If that's what they're really arguing, your Honor, they have a very strange notion of competition.

The bottom line is, they don't have to win the competitive battle. They don't have to drive us out of business. They don't

even have to knock our market share down in order to lose on fair use. The test is whether there's any non-trivial, any significant economic impact, and they have not come close to meeting that standard, not even close.

I don't think I could say it any better than Judge Leval did in the *Texaco* case in which he wrote, quote: "In order to prevail on the fourth factor, the copyright **[4-17]** owner is not required to demonstrate that it has been reduced to poverty by the defendant's copying." He continues: "The fact that the copyright owner is realizing rich profits from the exploitation of its copyrights despite the unauthorized copying has no logical tendency to prove that the secondary user's copying is not diminishing those profits. If the copyright owner would be receiving significantly higher revenue but for the defendant's uncompensated copying, the standard is satisfied regardless of revenues already being received." And they have failed to prove that but for their premise and infringement we would not be doing better, however it's measured, whether it's sales, whether it's market share, whether it's units, whether it's dollars.

If your Honor were to find fair use here on this record, I would respectfully submit having I believe fairly thoroughly canvassed the case law that this would be the first instance in history in which the use in the ultimately sold product of infringing material by a direct and vigorous competitor was excused as fair use. That's not what happened in *Sega*, that's not what happened in *Galoob*, and I've not found a case anywhere and they've not cited anywhere that kind of use by a direct competitor was excused as fair use. Thank you, your Honor.

THE COURT: Thank you.

[4-18] MR. GELHAAR: Your Honor, I will address the procedural aspects of the present posture of the motion and Mr. Reback will address the substantive aspects of that motion.

Mr. Gutman stated that Rule 52(c), he believes, is a proper vehicle to dispose of Borland's affirmative defense of fair use, and I am troubled by that. I have to also admit that I was

not entirely familiar with Rule 52(c), and after reading the Advisory Committee notes last night, it appears that it was indeed broadened to address claims by plaintiffs and defendants as opposed to the old 41(b) rule which only was against plaintiffs after the close of their case. But the rule was broadened as to both parties only insofar as the Court can enter judgment against a party concerning a claim, counterclaim, cross-claim or a third-party claim.

Now, reading the rule: "If during a trial without a jury a party has been fully heard with respect to an issue and the court finds against the party on that issue, the court may enter judgment as a matter of law against the party on any claim, counterclaim, cross-claim or third-party claim that cannot under controlling law be maintained or defeated without a favorable finding on that issue."

We're talking about an affirmative defense here. **[4-19]** Borland has not asserted any claim, and "claim" it seems to me must be read in the Rule 12 sense, like as if a failure to state a claim. It's not—the word "claim" is not used like claim, the assertion of something, but actually a claim which Borland has not—has not asserted in this case.

I could only find one case—

THE COURT: Well, let me just call your attention to another rule, 42(b). "The court, in furtherance of convenience or to avoid prejudice, or when separate trials will be conducive to expedition and economy, may order a separate trial of any claim, cross-claim, counterclaim or third-party claim, or of any separate issue or of any number of claims, cross-claims, counterclaims, third-party claims, or issues, always preserving inviolate the right of trial by jury."

Now, I think this would probably be inherently in the power of the court anyway if the rule didn't say that, but the rule says it. And I have ordered phased trials in this case, and I have included in my orders that the issue of fair use, whatever it is, would be tried in this proceeding and have given notice to Borland that it was being tried, and Borland has understood that and has offered its evidence on the subject.

Now, quite apart from whether there is any substantive difference as distinguished from change of **[4-20]** terminology between Rule 41(b) and new Rule 52(c) which became effective with the amendments that now apply to the trial of this case, I take it everybody agrees, quite apart from that I have said to you, I want this issue dealt with in a way that doesn't waste the resources of the parties.

Now, why should I be going beyond the present point before I hear your arguments on the merits as to what findings I should make on this issue and whether either as a matter of law or by making findings by a preponderance of the evidence on the evidence before me when Borland has had its opportunity and tells me, "We rest. We've offered everything we want to offer on the subject"? Why shouldn't I be deciding it now instead of wasting the resources of the Court and the parties and saying, "No, no, I can't do that till the end of the trial." Why?

MR. GELHAAR: If I might your Honor, if I can just address the question just preceding that one, and that is, after you read Rule 41(b)—42(b) rather, I see actually a difference in the language between 42(b) and 52(c). In fact, 42(b) preceded 52(c) and they're written differently. So it seems to me that—that one would infer that the drafters had a different intent. But you know what, your Honor—

THE COURT: Well—

MR. GELHAAR: —I'm not going to argue with **[4-21]** you.

THE COURT: Well, wait a minute. I'm putting to you a broader question now.

MR. GELHAAR: I think the Court actually—

THE COURT: Hold it. Suppose I don't have authority under 52(c). I have told you I had to invoke whatever authority I have to avoid a waste of the resources of the parties and to the public, and I want to hear the issue now. So do you say I don't have that power? If so, you've got to tell me I don't have it under 42(b) as well as that I don't have it under 51(c).

MR. GELHAAR: I don't believe the Court has the power under 52(c).

THE COURT: I'm sorry. I don't want the record to be confused by my misstatement of a rule number.

MR. GELHAAR: But the Court may, but I think not, have the power under 42(b), but I think the Court might have power under 12(f), because the Court may sua sponte move to strike a defense at any time during the proceeding.

THE COURT: Okay. All right. So if I have the power, I'm invoking it. I've told you that before and I'm telling you again right now.

MR. GELHAAR: But the Court has said in this case from the beginning that we're only going to try it once, and if it goes to the First Circuit, you want a full **[4-22]** record before the First Circuit so the First Circuit can decide all issues.

As I understand it now—I'm slipping a little bit into Mr. Reback's area, his argument, but there are cases in the Court of Appeals which say that the Court of Appeals upon considering the fair use defense can consider that defense de novo. Now, if Borland simply has its proof before the Court and the Court enters judgment on this issue, or does whatever it wants to do, strikes the affirmative defense, then the Court of Appeals has only heard Borland's side of this.

THE COURT: Well, of course. And so the only person who could complain then would not be Borland, who is trying to complain about it, but Lotus, who is not complaining.

MR. GELHAAR: But, your Honor, we're going against what you've stood for. I don't mean exactly that, but what you've said over and over again, we are only going to try it once.

THE COURT: Yes.

MR. GELHAAR: Why shouldn't I—

THE COURT: Yes, yes. But, Mr. Gelhaar, that doesn't mean that I carry that point to the extent that every time somebody says, "I've got an argument" I don't even evaluate that argument before I decide to spend several **[4-23]** days hearing evidence that would be relevant to it. I evaluate it. If my conclusion is when I evaluate it that once I make these findings there is no chance that this case will have to come back on this subject because I didn't let you hear Lotus' evidence—

you're complaining about Lotus not being allowed to offer additional evidence. What basis do you have for complaining about that?

MR. GELHAAR: But, your Honor, I think you're inferring that Lotus may well have some expansive proof that they would proffer on the fair use argument—

THE COURT: I don't care how much it is. An hour of this Court's time at $16,000 a day in taxpayers' cost is a waste of public resources as well as when I see the number of lawyers around this courtroom and how much it's costing the parties in attorney's fees, a considerable waste of private resources. One hour of time shouldn't be wasted here.

So now I come then to the bottom line proposition again. If I have the power to hear this issue and decide it now, I want to do it. It makes sense from every point of view, including another rule we haven't spoken about thus far, Rule 1: just, speedy and inexpensive determination of every action on the merits. That's what I'm aiming for.

So if I have the power, shouldn't I exercise it? Are you telling me no?

[4-24] MR. GELHAAR: I'm telling you, your Honor, that you should not exercise it unless you hear what Lotus has to say on it so as to avoid having to come back—

THE COURT: Tell me—

MR. GELHAAR: —down here a second time, assuming your Honor—

THE COURT: Cite to me any authority for the proposition that Borland has standing to complain because I'm not hearing Lotus.

MR. GELHAAR: Your Honor, the authority that I would offer to you is your prior hearings; and that we also intend—and we're not interposing this argument for delay, by any means.

THE COURT: Well, that's a misinterpretation of my prior hearings, but even if it were a correct interpretation, I always have the power to correct my prior errors before they get up to the Court of Appeals, as long as the case is still before me.

So you have misinterpreted what I have said in prior hearings. But even if you were correct, I would correct myself now and say I should exercise this power now.

MR. GELHAAR: Your Honor, Mr. Reback is pulling at my coat and that means either that I should not argue with you any further or that he is dying to get up to address you.

[4-25] (Laughter.)

THE COURT: Whichever of those reasons—

MR. GELHAAR: But in either event, I will submit to his coattail pulling and I will sit down.

THE COURT: All right. Thank you.

MR. REBACK: I just thought, your Honor, that I should be the one to continue to go forward because Mr. Gelhaar was kind enough to research this point and articulately present our opinion—position, rather. We do understand your Honor's point and the logic behind your Honor's point, so I'm going to proceed to address what I'll call the merits of our argument here. Before I do, I have one unrelated question with regard to the memorandum of yesterday, if I might just ask about that, sir.

At the top of page 2—and I may just be making too much of this, but at the top of page 2, the first sentence reads: "The parties have presented their evidence in a non-jury trial on the liability issues for the original complaint and are about to present evidence on liability issues for the supplemental complaint."

It just may be an issue of drafting here, sir, but the only thing that we had waived jury trial to were the three issues for Phase I, and that's all we put in evidence as to—at the last phase of trial.

THE COURT: Yes, we understand that.

[4-26] MR. REBACK: All right. Thank you, sir. So the record is clear here for the purpose of the argument—

THE COURT: You' re not suggesting that you're entitled to a jury trial on this issue we're now talking about.

MR. REBACK: The Key Reader issue, your Honor?

THE COURT: The fair use issue.

MR. REBACK: The fair use issue? No, sir. I'm not arguing about that. I'm arguing that I was entitled to a jury on the other—the other issues as to which I did not waive a jury.

THE COURT: Absolutely.

MR. REBACK: Okay. So the record is clear here, we raised the fair use defense as to the case in chief in our pretrial brief prior to the last session, docket number 311 at page 56. Your Honor was kind enough to permit me to argue the fair use issues last time and I will go over them briefly with particular emphasis on points that Lotus has made in briefing, and I will not repeat in—in totality the arguments that I made last time. I'm sure your Honor will consider those arguments in consideration of this issue.

I do want to point out as a preliminary that the statute, the case law, the legislative history indicates that the Section 107 issues are to be evaluated and weighed **[4-27]** by the Court, however the Court believes it's appropriate, and they are not an exhaustive list and the Court can take into consideration whatever it believes appropriate in assessing fair use in a particular situation, including this one. But the statute does lay out four factors, and I would like to address those factors, because Mr. Gutman has done so.

The first factor is the purpose and character of the use. And we have argued, your Honor, that virtually all uses have some commercial effect, but we've argued that that alone is not dispositive.

We argue that *Sega* and *Galoob* teach that that's not a binary decision, yes or no, that it's a matter of degree. Now, Lotus tries to distinguish *Sega* and *Galoob* by arguing that in those cases the defendant's product didn't compete directly with the plaintiff's product, and I want to say, your Honor, that I don't believe that that's correct, and let me start with the *Galoob* case as an example.

In the *Galoob* case, the defendant made a piece of hardware called the game genie and it allowed users of Nintendo software to enhance that software so that they didn't have to buy enhanced versions from Nintendo.

So, for example, to pick a name of one of these games, Super Mario Brothers I, if you had that, you could buy the game genie from Galoob, and you wouldn't have to buy **[4-28]** Super Mario Brothers II from Nintendo or Super Mario Brothers III, or any other games at an enhanced level from Nintendo.

Now, Nintendo didn't make a competing product with the game genie, but Nintendo argued that the game genie competed directly with the enhanced versions of its video games and indeed Nintendo argued it destroyed the market for those video games, and yet the court still found fair use. The game—

THE COURT: Well, are you suggesting, however, that the court found that the opposing argument was factually supported or wasn't it simply finding otherwise instead of accepting that supporting argument as factually supporting that—the argument that this would wipe out the market?

MR. REBACK: Well, the issue came up—I'm not sure I understand the question, sir.

THE COURT: Well, the way you stated the proposition, it sounded as if the court had accepted as factually correct the contention and wouldn't have to do that.

MR. REBACK: No. I understand your Honor's question. It wouldn't have to accept as factually correct the contention that the enhanced version would destroy—that the game genie would destroy the market for the **[4-29]** enhanced version, but it did accept as factually correct the fact that people bought the game genie instead of buying enhanced versions of Nintendo products. That—that was my only point. Nintendo code was present in the game genie, and—and so I would respectfully argue to the Court there are a number of dimensions of that case which bear consideration here, and I would argue the same is true of the *Sega* case.

In *Accolade*, the defendant in that case copied computer code software in the Sega console in order so that it could make games to compete with the Sega games, and it did that without getting a license from Sega. But fair use was still

found notwithstanding the fact that there was commercial impact on Sega's sales of the games.

Now, in both cases the defendants made money on the sale of their products and sold them for commercial purposes just as Borland sells its spreadsheet for commercial purposes, but we respectfully argue that that's just not dispositive. It's a matter of degree, and we would ask your Honor to take that into consideration.

Now, Lotus attempts to distinguish *Sega* by saying in *Sega* there was proof that what the defendant Accolade did was the only way to do something. There was proof or the court found that the only way *Accolade* could make a Sega compatible game was to copy the Sega code. Accolade could **[4-30]** clearly make—it did make other video games. They made games for Nintendo or Atar—or Commodore or Atari or other systems, but they couldn't make a Sega compatible system without the Sega code, much the same kind of argument we have here. You can't make a Lotus compatible spreadsheet without the Lotus menu command hierarchy. So we argue that those cases have dimensions which should—should help formulate the way we approach the problem here.

And we—we look to *Galoob*, because *Galoob* says that you should in the first instance in a case like this look to what the consumers do. Did they pay for the right to run their programs in the *Galoob* case? And we would ask the same question here. Did consumers pay for the right to run their macros on whatever spreadsheet they want to run macros on, and we—we would suggest they did pay for that and for that reason we think the case has applicability here.

The last point on the first factor, both *Galoob* and *Sega* emphasized in consideration of the first factor that the defendant performed substantial creative work of its own, and we would argue the same is true here on the record and the court has so found.

Now, I guess—I would say one more thing about the first factor. In *Sega*, the defendant clearly copied computer codes. Courts may disagree about whether the menu **[4-31]** com-

mand hierarchy is copyrightable or not, but there is no dispute in the case law as to whether computer code is copyrightable, yet *Sega* copied that—Accolade copied that and yet there was no infringement, there was still fair use, and the *Sega* court said that was because it was used to create new works, and we—we would respectfully ask the Court to consider that as well in the Court's consideration.

Now, the second issue—the second fair use issue in the statute is the nature of the work. And what I distill from Lotus' brief on this point is because we took copyrightable expression, Lotus argues in its brief there is no fair use, but if that was the case—if that were the case, there would never be fair use. Fair use only applies when copyrightable subject matter has been taken, and yet the court uses fair use in various cases to find no infringement. And I would argue, your Honor, the reason the court does that is because of the court's concern about expanding the copyright holder's monopoly, and that's what's going on in these cases and that's why the second factor is so important for functional works.

That's why *Sega* says it's the most important for functional works. It's not contradicting the Supreme Court. The legislative history says that these factors are to be evaluated according to the case and what's at issue, and *Sega* says when functional utilitarian works like computer **[4-32]** programs are at issue, the second factor is the most important. *Sega* says that utilitarian works like computer programs get only thin protection and that fair use should readily be found for computer programs and similar works, and, finally, the copyrightability—I'm sorry—the compatibility should be considered in evaluating the second fair use factor.

As I pointed out last time, and I won't go back into detail, in support of these propositions, *Sega*, *Baker vs. Selden* and *Altai*, many of the cases we've argued to this Court on the 102(b) proposition, and I—I do not intend to reargue them here but merely to ask the Court to take those cases into con-

sideration as the *Sega* court does in evaluating the second factor.

The third factor is the amount used. I think both parties are in agreement that the Court has made a decision. Certainly there's no issue as to trial or proof as to the amount used, it's already in the record, and there's only an argument as to which way the Court's holding tips the scales. The Court has made determinations as to how much was used. The Court has said, as Mr. Gutman pointed out, more than a trivial portion was used. We say that's neutral on that factor, but we leave that to the Court.

And so I come to the fourth factor, the **[4-33]** supplanting the demand test. The fourth factor deals with the effect on the market. And we argued there that when the Court is assessing the effect on the market, the standard is supplanting the demand for the original product. Is the market usurped by the defendant's product? And we cited in our brief *Sega* and *Galoob*, but that's just not the *Sega* and *Galoob* standard. It's the standard everywhere.

And I'd like to read into the record just a couple of citations from the Second Circuit also adopting the same standard. The first is *Consumers Union vs. General Signal Corporation*, 724 F.2d 1044, at 1050 through 1051, Second Circuit, 1984, and the more recent Second Circuit case, *Wright vs. Warner Books*, 953 F.2d 731 at 739. That's a Second Circuit case from 1991. And the point here is that when courts are talking about the standard, they say that the defendant's work may, their word, suppress the demand for the original work, but unless it usurps that demand there is no fair use. Now, Lotus cites primarily a—

THE COURT: Unfair use—

MR. REBACK: I beg your pardon, your Honor?

THE COURT: You said unless it usurps a demand there is no fair use?

MR. REBACK: You're quite right, your Honor. I misspoke. Unless it usurps—I appreciate that, sir. Unless it usurps the

demand, there can still be fair use **[4-34]** under the fourth standard.

Now, Lotus argues to the Court that a different standard should apply citing this case from New York. We would argue that that case—that Lotus is taking dicta from that case out of context. That case and the other case that Lotus cites for a different standard are cases in which what the defendant was distributing was an absolute identical copy top to bottom of the plaintiff's product. And what the court is saying in that situation, we don't have to go through a complicated analysis of the effect on demand of the original product, because the effect is obvious. Most assuredly, the District Court judge in New York did not intend to overrule the Second Circuit on the issue of standard.

So we think that the record is clear here under the supplant of the demand standard as to the effect of Borland's product on the market for Lotus' spreadsheet. There was absolutely no effect on market share. We have three separate data points, and they all come from Lotus. They don't come from us.

THE COURT: Let me —

MR. REBACK: The first data point already in the record—

THE COURT: Let me pause there. I don't understand what you're saying to me there.

[4-35] Isn't it correct that the question that I should be thinking about in this respect is not what was the Lotus market share at two different points of time, but what is the Lotus market share after this alleged fair use, and what would the Lotus market share have been if this alleged fair use had not—its conduct alleged to be a fair use had not occurred?

MR. REBACK: No, sir. I would disagree with that point. May I address that?

THE COURT: Well, now, this is essentially the same point Judge Leval was making, that whenever we are thinking about whether there has been harm or damage, we have to compare a factual circumstance, what did happen with what would

have happened if the alleged harm-causing event had not occurred.

Now, that's just fundamental law of causation all through the law, not just peculiar to this area. So you're telling me that I should disregard that fundamental principle that runs throughout the law of causation in all areas of the law and hold here that instead of comparing what has happened with what would have happened if the alleged harm-causing event had not occurred, I should just look at two different points of time as to what did happen.

MR. REBACK: Actually to make the record clear, it's three different points of time, beginning, **[4-36]** middle and end, but I will address your Honor's point.

Your Honor's characterization of my position—I don't want to be forced into this extreme notion that I'm arguing against the totality of the law of causation everywhere, but I am arguing against your Honor's application of that notion here.

THE COURT: Well, that's what I'd like to understand, because I don't understand why I should apply a different rule of causation here from the one that I apply in all the other areas of law that I have to judge.

MR. REBACK: Okay. Let me address that. The reason we have a supplant the demand test, the reason we look at what happened, what actually happened as opposed to what might have happened in the absence of the event is because of the—it goes to the fundamental purpose of copyright. The purpose of copyright is not for the benefit of the copyright holder. It's for the benefit of society. So we don't ask the question, is the copyright holder hurt in some general or remote sense.

THE COURT: Well, if we're thinking about the benefit to society, we think about what would have happened and what did happen, because the benefit to society is one that we can only determine by making this comparison. There are of course multiple other factors that affect what is the current market share and the comparison with what it was at **[4-37]** some point in time past.

MR. REBACK: Yes, sir.

THE COURT: You agree to that?

MR. REBACK: Well, I certainly agree that market share is determined by a lot of different things.

THE COURT: All right. So you're asking me to assume that none of all of those other things had any effect here—

MR. REBACK: No, sir.

THE COURT: —and only this one thing?

MR. REBACK: No, sir. All I'm saying to you is what was done did not supplant the demand. Let me complete the argument, if I might, on this point.

This goes back to Judge Breyer's very famous speech to a congressional subcommittee where he said that the purpose of copyright is to give no more monopoly than necessary to get the work out in the first instance. The point is not is the copyright holder hurt a little or is the copyright holder hurt a lot. The issue is, is the demand supplanted because the purpose of copyright is the proliferation of works. And if someone takes—

THE COURT: Well—well, wait a minute. But you see, you're missing another point. The issue is, is the public interest affected a little or not at all or a lot.

MR. REBACK: All right.

[4-38] THE COURT: And so that's the same kind of causation question.

MR. REBACK: I don't think so, sir, because in copyright law I would respectfully argue to the Court the purpose as articulated by Judge Breyer is that the public interest is served when there's just enough monopoly to get the product out in the first instance. And so if Lotus still holds a 70 percent share, and held it consistently notwithstanding these activities, they don't meet the fourth requirement. And that's our argument, sir.

And so I would respectfully disagree with the reading of Southern District of New York case, and I would argue that there is a specific reason why there's a supplant the demand test, and I would specifically argue we do not look to what

might have been. We look to what actually happened. And since what actually happened was no harm, then there's no foul. That's the argument.

THE COURT: But it's no harm from conduct, and harm from conduct has a connective element to it, a causation feature to it. So you're asking me to disregard that causation feature entirely, are you not?

MR. REBACK: I don't—I have difficulty adopting a question which asks about entirely, because my mind is just not as quick as the Court's is. My problem is—but I think I can agree with what you're basically **[4-39]** putting to me, which is that if Lotus—if Lotus would have had a hundred percent share of the market but for the conduct of Borland, that is not the test. If Lotus could have extracted monopoly profits on the sale of its product by charging consumers $5,000 for it but for the competition provided by Borland, that's not the test. And I go back again to the purpose of copyright. It doesn't confer that kind of monopoly.

THE COURT: Hold it. How is the—what's the statutory phrasing with respect to the fourth factor?

MR. REBACK: Give me a moment, your Honor, and I'll—

MR. GUTMAN: Your Honor, it's in Section 107. It reads, quote: "The effect of the use upon the potential market for or value of the copyrighted work."

THE COURT: Now, that doesn't say supplant.

MR. REBACK: No, sir. That's what the case law says—

THE COURT: Well—

MR. REBACK: —in interpreting what—you're quite right. Supplant is not in here. Neither of the issues—

THE COURT: Well, if that—

MR. REBACK: Neither is suppress.

THE COURT: Well, if that's what the case law **[4-40]** says, isn't the case law a gross departure from the reading of the statute?

MR. REBACK: No, sir. No, sir. The case law interprets the statute in light of the history and in the context of the copyright law as a whole. That's the reason the case law has come

to the position that it has. So I do I think that this presents squarely a legal proposition to the Court, and I certainly would agree that the—

THE COURT: Now, that is inconsistent with the argument you started out with that I must consider and weigh all four factors, no one is decisive.

MR. REBACK: I'm sorry, sir. I do not mean to say that the Court's conclusion on this factor is decisive, and I didn't mean to state that if indeed I did.

What I meant to say is, the argument about what further discovery or evidence is necessary on the fourth factor is disposed of by your Honor's conclusion as to what the fourth factor means legally. In other words, if the proposition is that the—what the fourth factor means is that all Lotus has to show is that there is some indirect harm, that a single sale was displaced, something like that, then most assuredly they can show that, most assuredly they can, and I would concede that point.

If, on the other hand, we are correct and the standard means what it says in the case law and it means **[4-41]** supplant the demand—

THE COURT: Now, notice that you have stated two extremes as my only options for interpreting that statute, no harm or just a little harm.

MR. REBACK: I didn't even get to state —

THE COURT: The point —

MR. REBACK: I didn't even get to state the second extreme.

THE COURT: —there's a whole intervening middle there that's open to me as a reasonable reading of the statute, isn't there?

MR. REBACK: Not if your Honor wishes to be consistent with the Second and Ninth Circuit case law.

THE COURT: As you read them.

MR. REBACK: Yes, sir, and as your Honor's reading of those cases, which is controlling in the court and not mine. That's certainly the case.

THE COURT: Well—

MR. REBACK: But I would respectfully argue to the Court that the Second Circuit and the Ninth Circuit set up a standard, and that standard is well accepted nationally, and that standard is suppress the demand. And any lesser standard I—I would respectfully argue to the Court is—

THE COURT: Well, wait a minute.

[4-42] MR. REBACK: —inconsistent with the totality of the copyright—

THE COURT: You said suppress. You've previously been saying supplant. Suppress is not the same as supplant.

MR. REBACK: Okay. Your Honor is correct. The test is not suppress. The test that I am arguing for as articulated by the Second Circuit and Ninth Circuit is supplant the demand. Sometimes I talk too rapidly and get ahead of myself, and I appreciate the correction, your Honor.

If your Honor please, just give me a moment to—

THE COURT: Surely.

MR. REBACK: I do want to go over the three data points so at least the record is clear, and I want to do that because at the time we originally filed our memorandum and opposition to Lotus' motion for directed verdict, they had designated as confidential the first data point. That's the reason your Honor—so I want to hand up to the Court now—because they have down-designated that, so I'd like to hand up to the Court now the unredacted version which is now of record so I can make the record on the three data points.

THE COURT: You may.

MR. REBACK: This replaces docket number 342.

[4-43] THE COURT: All right.

MR. REBACK: All right. The three data points.

THE COURT: Wait just a minute now, Mr. Reback.

MR. REBACK: I beg your pardon, sir.

THE COURT: Wait just a minute. Let the clerk mark it and let me look at it.

MR. REBACK: What we have tabbed on the side, your Honor, are the places in the brief where there was prior redaction.

MR. GUTMAN: Your Honor, might I ask if there's a copy for us from Mr. Reback?

MR. DETKIN: You've already been served with that, Mr. Gutman.

THE COURT: I'll let you see it in a moment, Mr. Gutman.

MR. GUTMAN: Thank you, your Honor.

(Pause - Court is reading.)

THE COURT: All right. You may examine the document.

MR. GUTMAN: Thank you, your Honor.

THE COURT: You may proceed.

MR. REBACK: Thank you very much, your Honor. Okay. Three data points.

[4-44] First data point: 1988, right after the original Quattro product was released, Lotus said in its speech to its own troops that they had a 70 percent market share and that if they maintained that 70 percent market share through the end of 1988 they would be, in their own words, home free. And they said at that speech that they shouldn't worry about who had the remaining 30 percent because that should not be of concern.

Second data point: October 17th, 1990, Mr. Manzi made a speech on the very same day that Quattro Pro 2.0 was released, and he said that Lotus still had a 70 percent market share.

Third data point: August 13th, 1990, two weeks after we pulled the Lotus compatible menus for the product, Lotus announced in a full-page ad in the Wall Street Journal, the New York Times, that they still had a 70 percent market share of units, to address Mr. Gutman's point, an 80 percent share of revenue. So I would respectfully argue to the Court that by their own admission they were home free.

Last point, your Honor. I've gone through the four factors. As we have discussed here today and as your Honor permitted me to discuss last time, this is—we are directing the Court

basically in equity on this point and the Court is free to consider whatever it deems appropriate, [4-45] and I would ask the Court to consider the implications of public policy here as I did last time, the fact that reasonable minds could differ on the protection of the menu command hierarchy, and as I will argue tomorrow in the Key Reader phase, that the menus are used only in a utilitarian sense and not in an expressive sense in the product. Thank you very much, your Honor.

THE COURT: Thank you.

MR. GUTMAN: Your Honor if I may just respond very briefly on Mr. Gelhaar's procedural point, I, too, paused on the point he did in reading 52(c) last evening and the fact that it doesn't expressly reference defenses, but I think in terms of the procedural context of the case now that needn't disturb the Court, and if I can have one second I'll explain why.

We have now tried both sides of every issue, both on our claims and on the defenses to our claims for Phase I. The only exception is this fair use defense. So if your Honor in analyzing everything that we did in February concludes that we have proven our claims in the original complaint and that they have failed to prove any of their affirmative defenses that were fully litigated to that original complaint, and the only thing left is fair use on which they have put in their case and rested and we have not put in a case, that we are precisely within the confines of [4-46] 52(c), because findings on that fair use point made by your Honor at that point would put your Honor in a position to resolve a claim, namely our claim on the affirmative—on the Phase I trial on the original complaint.

And although the rule is not cited, I believe that something like that is what—the *Coates-Freeman* case in rereading it last night, your Honor, it appeared, although your Honor did not cite Rule 52(c), but that was what was done there based on the proffers of proof by the parties. It was not a summary judgment decision, but your Honor entered what was described there as an interlocutory judgment, because I take it there were other issues that weren't resolved. So I suggest

that there's no difficulty even without resorting to these other rules, and if your Honor feels the need to resort to other rules, I would add Rule 16 to the list. I think it's clear that you can fit this within 52(c) without having to look at anything else. And I took it as very significant that Mr. Gelhaar did not quarrel with the proposition that the standard your Honor applies under 52(c) is preponderance of the evidence and not some higher standard.

Now if I may move briefly, very quickly to the merits arguments that Mr. Reback made. I think he's misstated *Sega*. He's misstated *Galoob*.

In *Galoob* what the game genie did was not what [4-47] further enhanced the program by Nintendo. The game genie speeds up Super Mario I, Super Mario II, Super Mario III. I could call one of my sons as an expert witness if that were necessary. It has nothing to do with replacing or supplanting or suppressing or influencing the needs of those other products. What the court found significant there was that Nintendo did not have any proof that they indeed had foresworn ever going into this enhancement market.

And one important distinction between that case and this is because the fair use analysis in *Galoob* was not on Galoob which made the game genie. It was on the users in their homes who bought it and used it to affect the Nintendo product. That was not a commercial use and, therefore, unlike Borland here, which has the burden on the fourth factor, Nintendo had the burden on the fourth factor because it was not a commercial use. If it's a commercial use under the first factor, that puts the burden under the fourth, and in the *Galoob* case, the burden—

THE COURT: What difference does it make where the burden is? The burden only decides the issue where the court or fact finder is in balance and can't make a finding by a preponderance of the evidence.

MR. GUTMAN: And I—and I would certainly hope that that is not relevant here, your Honor, so let me move—let me move to *Sega*.

[4-48] In *Sega* the court—first of all, this was only intermediate copying. There was no copying reflected in the final product. The Ninth Circuit made that abundantly clear, and as they were analyzing all the factors that Mr. Reback's referred to, they made that clear, and if one reads it out of that context, one is seriously misreading *Sega*.

Second, the court in *Sega* found that the commercial aspect of the use can best be described as of minimal significance. This is at the top of page 1523. There is no—I would submit that no such finding could be made on the record here.

Finally, this—this notion that there is a supplanting standard—I have read and reread all of these cases, your Honor, and I can't find it. The only case that talks about supplanting is *Harper & Row* in the specific context of the Time magazine use, supplanting—I'm sorry, the Nation magazine use—supplanting the prepublication rights which Time magazine had paid for and which resulted in Time magazine getting $12,500 back, which I suggest is probably less than the damage they do us—do to us every day they infringe. Because of the prepublication, that was supplanted. But neither *Harper & Row* nor any other case, not in the Ninth Circuit, not in the Second Circuit, nowhere, says that supplanting is the standard that must be met.

[4-49] And, again, if I can read from *Sega*, same page, page 1523, after quoting *Sony* for the proposition that we must of course inquire whether if the challenged use should become widespread it would adversely affect the potential market for the copyrighted work, they quote—the Ninth Circuit quotes their own decision in the *Hustler* case, quote: "by diminishing potential sales, interfering with marketability, or usurping the market." Interfering—and this is—interfering with marketability, diminishing potential sales or usurping the market. That's how the Ninth Circuit states the test in the *Sega* case. So I don't know where the supplanting comes from, but it is certainly not the test. It is certainly not the minimum that must be established in any court anywhere.

Last point on market share. If I may just suggest to the Court a hypothetical which I find helpful in dealing with the argument by my brother that if our market share was 70 percent before they started infringing and our DOS market share—and that's a detail he left out of the ad—was still 70 percent as of this summer, let us take a hypothetical which is probably not very far from the truth.

Let us assume that in 1987 when we had that 70 percent market share we were selling Lotus 1-2-3 on average for $300 a copy, and let us assume that in order to still have a market share several years later of 70 percent, after **[4-50]** having had to compete during that period of time with somebody selling an infringing copy of our work for a much lower price, we are now still selling 70 percent, but at an average price of a hundred dollars a copy or a hundred fifteen or a hundred twenty, or whatever it turns out we could prove it was. I would suggest to your Honor that by any measure that is a serious injury, a serious injury, and it is something that this superficial approach to market share numbers completely overlooks. Thank you.

MR. REBACK: Your Honor, just one point. I would like to read into the record a quote from *Sega*, 977 F.2d, 1253—I'm sorry, at page 1523. "A use"—quote: "A use that effectively usurps the market for the copyrighted work by supplanting that work"—and I'm going to go out of the quotation now. The court says that that may not be the fair use. Then, quote: "The same consequences do not and could not attach to a use which simply enables the copier to enter the market for works of the same type as the copied work."

We have some other quotations from cases using supplant on page 20 of our brief, your Honor, docket number 342.

THE COURT: All right. Let me speak first briefly about the interpretation of Rule 52(c).

I confess that as a member of the Standing **[4-51]** Committee on Rules of Practice and Procedure of the Judicial Conference I did not notice this curiosity of the drafting as it went through the process before it was adopted effective

December 1st, 1991, but I do note even now that the rule starts out with a phrase that was sort of lost in our discussion earlier today. The starting phrase is, quote: "If during a trial without a jury a party has been fully heard with respect to an issue."

Now, it is true that "issue" was dropped out in the later language of the rule. But even apart from the other sources of authority to which we have referred in our colloquy earlier today, it seems to me I should read Rule 52(c) alone as authorizing me to hear and decide as fact finder making partial findings an issue at this point and tell everybody we're not going to hear further evidence and argument on this issue even though I may not be in position to order judgment on a claim or defense, et cetera, until I have completed the rest of this trial.

So for that reason, in addition to all the other sources of authority, it seems clear to me that it's entirely appropriate for me and would be appropriate for me to be doing it at my own initiative, and I am giving you notice that that's what I would be doing. It's also appropriate to treat it, it seems to me, as being before me for ruling under the motion by Lotus when Borland has **[4-52]** presented all the evidence it proposes to offer on this issue. So I proceed to decide this issue at the present time.

I find by a preponderance of the evidence that Borland's use of copyrightable elements of Lotus 1-2-3 was not a fair use. Now, that of course is my ultimate finding of fact by a preponderance of the evidence. I reach that after consideration of all of the factors, and I will speak further in stating some supplemental findings that support that ultimate finding.

Before I go to those supplemental findings, let me say also with respect to an issue of law that I accept Borland's point that a factor analysis in weighing is required under the terms of the statute, 17 United States Code, Section 107, and therefore I am not making any legal or factual determination that any single factor is alone controlling. Even if that might be a possibility with respect to some of these factors, I need not

address that question in the present circumstances because I at least find that there is no single factor here which would be decisive in favor of Borland, and having made that determination I need not address the question whether there is any single factor that might have been decisive in favor of Lotus. I don't need to do that because under the weighing and comparison of the effect of all of the **[4-53]** factors—and I accept the point also that Borland has argued that this may not be an exclusive rule. When I accept the obligation to weigh and evaluate all the factors, then I come to the finding that I have stated at the outset, that by a preponderance of the evidence, that Borland's use of copyrightable elements of Lotus 1-2-3 was not a fair use. Now let me proceed with some of the supplemental findings and the order of the factors as stated in the statute and in the precedents.

The purpose or character of the use is the factor I address first, and I find by a preponderance of the evidence, again, that the purpose and character of the use by Borland in this circumstance was essentially and primarily commercial and competitive. This is not a case of use for scholarship or research or for internal or in-house use only. It is not a use only for the purpose of understanding the ideas in order to make use of the ideas in another work. It was primarily aimed at commercial and competitive objectives of entering the market with a product that would compete very successfully with Lotus 1-2-3 in its initial and varying versions as time went on.

Quite important to that objective was the development of macro compatibility as one of the factors. In order to reach in its marketing, the marketing of Borland, not only previously uncommitted potential customers **[4-54]** but also the existing group, large numbers of owners of versions of Lotus 1-2-3 who had developed their own macros and files and data and had thus put an investment into what they had, that would be sacrificed in a sense if they bought a new product that could not make use of the existing in-house products of the customer built upon 1-2-3 in the development of both macros and

files in which the macros were used. So the purpose and character of the use as a factor in the factor analysis weighs very heavily against Borland.

Now I turn to the second factor, the nature of the work. In this respect I take account of the rulings with respect to nature of the work that I have already made as a matter of law with respect to the issues that I have decided on the motions for summary judgment, and under those determinations the Lotus 1-2-3 work as to which copyrightability and copyright infringement is being claimed in this case had expressive and not merely functional or utilitarian characteristics that are copyrightable, and those were substantial.

Now, at this point I take note that for my previous purposes of making the summary judgment ruling I did determine as a matter of law that the expressive elements constituted more than—the expressive elements that were used by Borland were more than a trivial portion. **[4-55]** At this point I have yet to make rulings in this trial with respect to some further issues as to how extensive the use by Borland of expressive elements came to be. I need not make that determination at this point precisely in order to decide the present issues before me with respect to fair use, but I do take note of the fact that the use that has been made goes beyond just, quote, more than a trivial portion, unquote, and take that into account as one of the additional factors that I am considering with respect to the fair use issue.

Now I turn to the third element, the amount and substantiality of taking, and the point that I have just made is of course relevant here as well. The determinations that I have made as a matter of law in the previous rulings, when I take them into account along with all of the other factors in this analysis, are sufficient to support the ultimate finding that I have made, and for that reason I do not address that issue further at this time. I will be making further findings after we've completed all the evidence before me in the remainder of the trial on that issue for other purposes rather than simply this purpose of deciding the fair use question.

As to the fourth factor, I first take note of a legal ruling. I do not read the cases, and in particular *Sega Enterprises vs. Accolade* by the Ninth Circuit in 1993 **[4-56]** and *Lewis Galoob Toys vs. Nintendo* by the Ninth Circuit in 1992, as establishing a legal standard of supplanting the copyrighted product in the market. Those opinions contain clear indications that if there was supplanting found, then of course that would weigh very, very heavily in favor of a determination that the use was not fair. They do not say by any means that supplanting is essential even to a determination that this fourth factor weighs against fair use. And I would be deeply concerned if they did say that as to whether I should follow it in light of what appears to me to be a deviation from an appropriate reading of the congressional directive. We, after all, as judges must follow that congressional directive in 17 United States Code, Section 107, with respect to the way this fourth factor is phrased.

Also, there is as I read the statute and the precedents built into this fourth factor a causation requirement, and that means to me that as I apply that standard to the facts of this case, that I must find by a preponderance of the evidence that this use, the conduct of Lotus in relation to this use, does have and has had a harmful effect on the Lotus market share and the appropriate—

MR. REBACK: I think you misspoke here. The conduct of Borland I believe you meant to say.

[4-57] THE COURT: I beg your pardon?

MR. REBACK: I think you just misspoke. I think you meant to say the conduct of Borland.

THE COURT: Did I say the conduct of Lotus?

MR. REBACK: Yes, you did, your Honor.

THE COURT: Well, if I did, thank you for correcting me. The conduct of Borland is what I'm talking about did have a substantial impact on the market share that Lotus would have had.

And I take note also of the point that Mr. Gutman has made which I think is correct, that it's not even market share that is

the final decisive point as to whether there's been some harm, because if the market share has been maintained only in a competitive situation that causes price reduction, that still has had a harmful effect.

Now, when I put all of these factors together, then, as I have stated at the outset and I repeat finally, I find by a preponderance of the evidence that Borland's use of copyrightable elements of Lotus 1-2-3 was not a fair use.

Now, that completes my statement of findings and conclusions on this issue. I invite the parties right now, both of you, to tell me if there's anything that I have stated that you wish to call to my attention as a—as you think an error or misunderstanding. I'm not asking for reargument, but, for example, you've just kindly called my **[4-58]** attention to a misstatement I made with respect to the name of the parties. If I've done anything else of that kind in the course of these statements, of course I invite you to call it to my attention now. Also, if either of you thinks I have not addressed some issue that is essential, I invite you to bring it to my attention now.

MR. GUTMAN: No exceptions, your Honor, and I can't think of any additional findings we would need.

MR. REBACK: Your Honor, I did listen carefully and I have nothing to state at this time as well.

Mr. Gelhaar has pointed out to me that under the judgment rule, I have ten days after the entry to file an amendment. I just wanted to see the transcript if I may do so, but nothing comes to my mind immediately, sir.

THE COURT: Certainly.

All right. Now, let me confer with you about what we do next and the answer to that may also determine whether I take a 20-minute recess at this point or later.

What's next on our order of proceedings?

MR. GUTMAN: Again, your Honor, my assumption was that once we disposed of the issue we've just dealt with, we would be proceeding with the Phase II trial pursuant to the stipulation the parties signed; that is, the Key Reader

supplemental complaint trial. I believe the only—strike "believe."

* * *

[4-69] MR. GUTMAN: Excuse me. He will explain that in the demonstration—

MR. REBACK: I beg your pardon.

MR. GUTMAN: —and please wait.

DIRECT EXAMINATION

BY MR. GUTMAN:

A. Okay. I'm all set.

Q. Thank you. Mr. Roshfeld, I realize you've already been examined, but could you state your full name for the record?

A. Larry Mark Roshfeld.

Q. And where are you employed, sir?

A. I work at Lotus Development Corporation.

Q. And in what capacity?

A. My title is Senior Product Manager for spreadsheets.

Q. And how long have you been at Lotus?

A. It will be nine years in July.

Q. Now, did you prepare a demonstration file for use in your demonstration today?

A. Yes, I did.

Q. And did you also prepare extra copies of it?

A. Yes, I did.

Q. Could I show you—

MR. GUTMAN: May I approach the witness, your Honor?

[4-70] THE COURT: You may.

A. Yes.

Q. Are these in fact three copies of the demo file that you prepared?

A. That's correct.

Q. And they're identical to the one that's now in the machine?

A. That's correct.

MR. GUTMAN: All right. Your Honor, I would like to mark—so the Court may have this available for the Court's use in preparing a decision, I'd like to mark as the next exhibit in order this—

THE COURT: What is the next exhibit in order?

MR. GUTMAN: I am hoping that as I approach someone more knowledgeable than I, I will provide that answer.

MR. REBACK: Excuse me, your Honor. Just give us one second, please.

THE COURT: Yes.

MR. REBACK: It's the first time we've seen this.

THE COURT: All right.

(Pause.)

MR. GUTMAN: I believe it's Exhibit 49, your Honor.

[4-71] MR. REBACK: I'm sorry. Forty-nine?

MR. GUTMAN: I believe that would be the next number. According to the indices of exhibits that would be the next number.

THE COURT: All right. It may be so marked.

BY MR. GUTMAN:

Q. Now, Mr. Roshfeld, could you tell the Court, please, which products you're going to be demonstrating?

A. Yeah, I'll be demonstrating 1-2-3 release 2.01; Quattro Pro for DOS release 3, Quattro Pro for DOS release 4.01, or at least that's what I was told to use—to refer to that product, which is Quattro Pro for DOS release 4 manufactured and sold subsequent to August of 1992; and Quattro Pro for Windows. And I'll be running all of these products from Microsoft Windows version 3.1.

MR. REBACK: Your Honor, to avoid interrupting the presentation I do need to state an objection for the record here to preserve the objection—

THE COURT: All right.

MR. REBACK: —if I may. The witness has testified that the only Lotus product he's going to demonstrate is Lotus version 2.01. Your Honor will recall we previously argued, and I don't intend to bore the Court with that argument again, that we're not charged with infringement of that product. There is an earlier **[4-72]** stipulation in this case, but that stipulation was entered into prior to the Key Reader phase and does not affect the Key Reader phase of the case.

I know from the record that there is some difference between 2.0 and 2.01 that affects the macro compatibility. I've not had the opportunity to take discovery on that because the Court precluded discovery until after release versions of Lotus 1-2-3.

So I just want to state for the record that's not what we're charged with infringement of, and I don't know the difference between that and 2.0 insofar as macro capability is concerned.

MR. GUTMAN: First of all—

MR. REBACK: Oh, excuse me. One other point. We did ask him to load 2.0 and to demonstrate that, but apparently they were unable to do so.

MR. GUTMAN: Nobody had a copy of 2.0 that could be loaded, your Honor. I believe that the problem and we have stipulated and there was nothing in that stipulation that restricted it to Phase I of the trial as opposed to the entire case that there were no material differences between 2.0 and 2.01, number one.

Number two: Both of us in the previous demonstrations we did, which were videotaped in connection with the summary judgment motion, used 2.01. So if there's **[4-73]** something wrong with using 2.01 which is readily available and easily loaded and stored on a hard drive and easy to use for the demonstration, it's a little late in the day for Mr. Reback to be raising that point and it's a bit unseemingly in light of his stipulation.

Three: He was never foreclosed from any discovery with respect to 2.01. He had full discovery on that subject; and to the extent that there were issues as to later released versions of 1-2-3, they were releases after 2.01.

MR. REBACK: Let me just respond briefly. We have objected at each point when they've attempted to demonstrate version 2.01 as opposed to version 2.0, so I think I'm protected on that. There's no inconsistency here. We've always raised that.

There is a material difference between the issue as it pertained to the rest of the case and the issue as it pertains to the Key Reader.

The stipulation was entered on October 28th, 1991 before the Key Reader was in the case and says there are no material differences in the user interface of the two versions that would have any effect on the issues in this case. That was before the Key Reader.

We're now trying the issue of macro compatibility. I do believe there is some difference in **[4-74]** macro compatibility between version 2.0 and 2.01. I just don't know what it is.

THE COURT: Well, I will overrule the objection. It seems to me it is quite analogous to an objection that is made when a photograph is being introduced and there's some difference between the photograph introduced and the circumstances that are relevant to the case; and the common practice is to receive it in evidence with permission for the opposing side to show me what any differences could be that would make any difference.

Now, in light of the explanation, which I take it you're not challenging that a version of 2.0 that could be loaded for this demonstration was not available, if you've got one, then you may put it on at some point and we'll see if it makes any difference.

But I am overruling the objection without prejudice. This is an objection to receiving it into evidence. If you have anything to show me in cross-examination or by way of counter-demonstration to support your hypothesis that there's something here that makes a difference, you'll be free to do it.

MR. REBACK: All right. I understand that, sir. And we will show you deposition testimony at the relevant time. But I have to say without having had discovery on the issue, it's

going to limit my ability to **[4-75]** show whatever I need to show.

THE COURT: Well, now there is a flat contradiction between what you're saying to me with respect to whether you've had discovery on this issue. Mr. Gutman is telling me that you did on 2.0 and 2.1, and it's only later versions to which there was a preclusion on discovery.

Now what's the basis for telling me you were precluded from discovery with respect to 2.1?

MR. REBACK: Your Honor, I thought that the order your Honor entered a couple of years ago precluded me from taking discovery of any version of Lotus 1-2-3 subsequent to what I was charged with infringing in the complaint.

THE COURT: Mr. Reback, I would never rule on the basis of my memory of my order two years ago. It was an order that's on the record somewhere. So if you contend that there was something about my order that precluded you from discovery with respect to 2.1, show it to me.

MR. REBACK: Okay.

THE COURT: And I'll permit you at the time to do that during a recess and bring it to my attention.

MR. REBACK: Thank you, sir.

THE COURT: All right. You may proceed.

MR. GUTMAN: Thank you, your Honor.

BY MR. GUTMAN:

[4-76] Q. Now, in setting up these programs that you're going to demonstrate just a few moments ago, you changed the Quattro products from WYSIWYG mode to character mode. Is that correct?

A. That's correct.

Q. Could you explain to the Court why you did that?

A. The version of 1-2-3 that I'm showing 2.01 only displays in character mode. The versions of Quattro Pro for DOS, Quattro Pro 3 and 4.01 default to displaying in their WYSIWYG mode which is graphical font mode. The difference is that in 1-2-3, as you can see on the screen, it shows 20

lines; and Quattro Pro in their WYSIWYG mode it shows like 22 lines, I believe.

The problem was—for me was that when I set up this demo file so I could show one screen at a time, the breaks between the screens got changed because the screen went from 20 lines to 22 lines. It doesn't really affect anything that I'm showing you.

If you'd like I'd be glad to go back into the WYSIWYG mode. The only difference is it means that something that I would not have wanted to show in a demo until later on will now be visible because of the problem of these additional lines affecting the break on the screen.

Q. Thank you, Mr. Roshfeld. Now, the program showing now is Lotus 1-2-3?

[4-77] A. That's correct, Lotus 2.01.

Q. And this is your demonstration file?

A. That's correct.

Q. Okay. Now am I right in understanding that the items in column B in cells one, three and five are macros that you prepared in advance?

A. That's correct.

Q. Just so we're clear on how a macro is prepared. Could you please erase the first two and recreate them; and since our only record of your demonstration today is going to be the transcript the Court Reporter prepares I would ask that, to the extent you can, you speak out loud the various keystrokes you're undertaking so we'll have a record.

A. I will try to do that.

So the first thing I'm going to do is erase the cells B1, B2 and B3 to erase the macros that are put in them. To do that I'll hit the slash key, bring up the 1-2-3 menu, choose the range command, choose the erase command, and then highlight those three cells, excuse me, highlight those three cells and press enter, and that just erases the content of those cells.

THE COURT: Wait a minute. You only highlighted two of them, didn't you?

THE WITNESS: I highlighted the blank cell in between, your Honor.

[4-78] THE COURT: Oh, I see. All right. Okay.

BY MR. GUTMAN:

Q. Okay. Now, what's the purpose of the first macro, the macro labeled A?

A. The first macro works through the menus in 1-2-3 and chooses first the slash key, brings up the range—excuse me, brings up the menu itself. The letter R brings up the range menu, the letter F chooses the format, the submenu. C chooses currency, and then it accepts the default for two decimal places, and then it formats the current cell, so it's just that a macro, that it automates the process of formatting a cell to be two decimal places with the dollar sign in front of it.

Q. Now could you type in that macro showing the Court how it is created?

A. The first thing I need to do is put in a label prefix. Because if I simply type the slash key, it doesn't go in as an entry to the cell. It actually brings up the menu.

And so first I type in a label prefix, and then from then on what I'm typing is essentially the label. So I type in the slash key which will bring up the menu in the macro it's on, the letter R to bring up the range menu, the F for format, the C for currency. At this point if I were doing it manually, I would hit the return key to accept the default of two decimal places.

[4-79] In macro, the way you represent the return key is by using the tilde symbol. And then at this point if I'm working through the menus manually, I would be prompted for the range, and the default would be the current cell.

In this case I'll accept that, but then another tilde in which again represents another enter, and that would complete that command sequence. So since there's no subsequent commands on that line or any following lines, the macro would cease execution at that point after it's formatted the cell.

Q. Now, could you recreate macro B explaining what you're doing?

A. If I go down to cell B3, which is where I had previously had macro B, again, I put in the label prefix so that I can enter in the rest of the macro, put the slash key in that will bring up the menu, the letter W for worksheet, the letter G which will choose from the worksheet submenu, the choice global, the letter F for format.

So what I'm doing in this macro is changing rather than the format of a single cell the global format for the entire current worksheet. I'll choose C again for currency, and then accept the default for two decimal places again by putting a tilde and representing enter.

The difference between this macro—the reason I only need one tilde versus the other where I needed two was **[4-80]** that the other command only worked on a portion of the spreadsheet so I needed to specify which portion. This is a global command, so I don't need to specify a portion because by default it operates on the entire worksheet; and then to finish the macro I think just press enter and that enters into the cell as a label.

Q. Now I notice that you didn't name either the A macro or the B macro. Why is that?

A. Well, I had—the reason I didn't name either the A macro or the B macro is that they had been named previously.

If I wanted to name those macros so that 1-2-3 knew that it were macros, that's done by assigning them a particular type of arrangement, and I've already done that. But to show you how I would do it again, I would hit slash key to bring up the menu, choose the range option from that menu, and then choose from the range menu, name, and then create, and what I want to do is create a macro name.

Macro names by default in 1-2-3 are starting with the back slash character followed by a single letter. So in this case if I wanted to name the A macro, I would call it back slash A, press enter, and then specify the range which contains that macro, in this case cell B one. So I'd press enter again, and

that's what I had done previously in this demo file to name these macros.

Q. Now, again, describing what you're doing for the **[4-81]** record, could you please demonstrate to the Court how macros A, B and—well, let's make it A and B for now work.

A. The first thing I'm going to do is move my cell pointer over to cell C9. As you can see on the screen, there's a series of columns of numbers. So there's five columns, each column having six numbers in it. And the first thing I'm going to do is run that back slash A macro which will point out the current cells which contains the value 456 as currency to that macro.

To execute a macro, I first press down the Alt key, A-L-T key, on the keyboard, and press the letter that I had assigned to that macro; so in this case it's the macro back slash A, so I press the letter A.

So Alt A, and it went very quickly, but this is a fast computer. But what the macro did is passed those same key strokes to 1-2-3 that I would have done manually so the first keystroke brought up the menu.

In this case the slash brought up the menu, R chose range, F chose format, C chose currency. At that point I was prompted for the number of decimal places, and then I pressed enter. In fact, why don't I just show you what the menus looked like.

Slash key brings up the menu. It would then choose range by pressing or—excuse me, bypassing the letter R to the menus, F chooses format, C would choose the **[4-82]** currency option. At this point it defaults to two decimal places, so I just accepted that by pressing enter. And then it defaults to the current cell for the destination for that formatting, and again I accepted that by pressing enter. So I just manually repeated the same process that the macro had done.

Q. Thank you. Now could you show us macro B in operation?

A. As I had mentioned earlier, macro B will globally format the entire worksheet as currency, two decimal places, so

the rest of the numbers on the screen will display with the dollar sign and two decimal places.

To run the back slash B macro, again, I hold down the Alt key and press the letter B.

Q. Now I see you've got a column of asterisks over in the G column. Why is that?

A. The numbers in the G column were larger. They were in the tens of thousands, I believe; and so there wasn't enough room in the default column width to display the number with two decimal places and preceding dollar sign.

So in 1-2-3 we display asterisks there to signify that there's contents there, but there's not enough room to display it accurately.

Q. Now, if my hunch is right, macro C would solve the problem?

[4-83] A. Yeah. Macro C will change the column width of the current column to a width of 15. The default in 1-2-3 is nine characters wide, and through experimentation I determined that 15 was wide enough to adequately display this.

So what this macro will do is first bring up the menu. To depress the macro, the W will choose work sheet, the C will choose column, the S will choose set width. It then pauses and prompts the user for the width to set. We will enter in the numbers one and five for 15, and then the tilde accepts that.

So to run this macro I press Alt, and the letter C, and the column is widened to a width of 15. You can see in the top left corner of the screen next to the cell address where it says G9 colon, it displays W15 within square braces and that signifies the width of the column.

Q. Okay. Now, are you familiar with something called interactive macros?

A. Yes, I am.

Q. Okay. Could you tell the Court, please, what an interactive macro is?

A. In the macros that I just demonstrated they were what we would call hard coded. There was no variation within the

structure of the macro itself. It did a particular task from beginning to end.

[4-84] Interactive macros are intended to allow the users to provide options in the midst of the execution of macro. So, for example, if I were writing a macro for either my own use or for someone else's use, I wanted them to be able to specify in the midst of the macro the number of decimal places they wanted to display, I could design a macro that paused at that point, allowed the user to enter whatever number of decimal places they choose, press the enter key, and then the macro key would continue execution.

Q. Do you have some prepared?

A. Yes. If I page down now, you'll see that the screen now shows rows 21 through 40, the same columns, so I just simply press the page down key to move down on the screen to a subsequent series of roles, and here I have four different macros.

MR. REBACK: Excuse me, sir. Are they also on this disk that we've gotten a copy of?

THE WITNESS: Yes. Everything that I'm showing you comes from a single file called demo.WK1.

MR. REBACK: Which is on this disk?

THE WITNESS: Yes. And the file demo.WK1 is on that disk.

MR. REBACK: Thank you.

BY MR. GUTMAN:

Q. Okay. Now, could you tell us, please, what macro D **[4-85]** does?

A. I'm going to move my curse back over to cell C29. Macro D is similar to the previous macro I had shown that did formatting of currency to two decimal places.

The difference in this macro is that it prompts the user in two different places during the execution of the macro for feedback. So it does flash to bring up the menu, R for range, F for format, C for currency.

At that point we instruct the macro to pause by using a particular command in the 1-2-3 macro language which is those squiggly braces with a question mark between them followed by a tilde which accepts the user's input.

So at that point it pauses and allows the user to determine how many decimal places they wish to format, but it is hard coded for currency, so it will always do currency, but it allows you to specify the number of decimal places; and after you specify the decimal places they can then specify the range to effect.

Q. Thank you. Could you show us how macro D works, please?

A. I'll run that macro by pressing Alt D. The macro works its way through the menu, pauses at the point where I'm prompted for the number of decimal places.

In this case I'll choose zero decimal places, press enter to accept that. The macro then can cease **[4-86]** operating, gets to the next step, which is a pause for the range to format, and I'll use my cursor keys to highlight a range of three cells by three cells; and now when I press enter, you can see that the change on the screen--those nine values are now displayed as currency with zero decimal places.

Q. Now could you describe macro E for us, please?

A. Macro E is an extension of what we just saw in macro D. It's interactory both for the type of format as well as for the number of decimal places and the location.

So in macro E, the slash key brings up the menu, R chooses range, F chooses format. At that point the user is left in the format submenu. They can then point to whatever particular format type they choose. When they press enter, they're then prompted to the number of decimal places, and when they press enter again, they're prompted for the location.

Q. Could you show us how it works, please?

A. To run that macro, choose Alt E. In this case rather than just defaulting to currency I can choose whichever one I'd like.

In this case I'll choose scientific format, instead of exponential format. I'll put in just for variety one decimal place, and I'll highlight a two-by-two range of cells. And now when I press enter, those four cells are **[4-87]** formatted as exponential or scientific notation.

Q. Thank you. Now could you describe, please, macro F.

A. Macro F is also an interactive macro. It brings up the menu, chooses worksheet, and then chooses insert, and at that point the user can specify what they want to insert.

If I show you that manually on the menu, it's the equivalent of hitting the slash key to get a menu, worksheet, and then insert, and then at that point the macro pauses to allow me to choose either column or row. And I'll just escape out of the menu using the escape key so I don't actually generate that operation.

Q. Now could you show us how macro F works?

A. Pull down the Alt key, press the letter F. It asks me for column or row. I will specify row. And what it will do is insert a row, prompt me for the location where I want that row inserted, and I'll say at the current location, and by default it inserts a row above the current cursor position.

Q. Now could you describe macro G, please.

A. Macro G is similar to macro F in that it brings me to the worksheet insert menu. But at this point rather than allowing me to choose between column and row the R in that macro specifies row, and then all I have left to choose at that point is the location where I want the row inserted.

Q. Okay. Could you show us how that operates, please.

[4-88] A. Hold down Alt and the letter G. And now it's gone beyond that column or row choice, and it's gone straight to the point where it prompts me to enter row insert range. And if I press enter at this point, it inserts another row so the effect visually on the screen was to move that range of data down away from the macros.

Q. Okay. Thank you. Mr. Roshfeld, I'd now like you to move the demonstration to Quattro Pro release 3.0.

A. What I'm going to do since I'm running under Windows I can switch between different products running at the same time. To move between the products I'll take the end of the hot key, in Windows which is the Alt tab key, which allows me to cycle through all the different applications that are loaded in memory until I get to the one I'm interested in at this point, which is Quattro Pro 3.

So what you see on the screen right now is Quattro Pro 3. I have retrieved that same demo.WK1 file from the same floppy that I used to retrieve the file 1-2-3.01. And as you can see I'm in the character mode in Quattro Pro in order to display the same number of rows on the screen as I had in 1-2-3.

Q. Okay. And am I right in understanding that this file, this demo file, was prepared using Lotus 1-2-3?

A. That's correct.

Q. Now, could you show us, please, how macro A works in **[4-89]** Quattro Pro 3.0?

A. Okay. If I wanted to run this macro I would do the same thing as I did in 1-2-3. I would hold down the Alt key and hit the letter A, which is the back slash A macro, and something very different happened than what had happened in 1-2-3.

Q. Right. Okay. Can you escape from that?

A. (Complies.)

No, I have to reopen the file. It appears that what had happened is that in the course of processing those keystrokes for some reason removed that file from memory, so again I'll go out to the A drive which is my floppy drive and retrieve that demo.WK1 column.

Q. Now you were using the Quattro native menu in Quattro Pro 3.0 at that point, correct?

A. Yes. The native menus in Quattro Pro 3.0 are currently on the screen.

Q. Okay. Now could you turn on the macro Key Reader, please, and try it again.

A. To turn on the macro Key Reader in Quattro Pro 3, I hit the slash key to access the menu, I move my cursor over to the tools menu, press enter.

Now I want to select the macro submenu, so I'll press enter, since macro is already highlighted; and then if I move my cursor down to the bottom of that list, you'll see **[4-90]** Key Reader. I press enter, and it now says the default was no, however, it had been set previously, and I'll change that to yes. So I've just turned on the macro Key Reader feature in Quattro Pro 3.

Q. And now could you try macro A, again, please?

A. Again, I'll try macro A by holding down the Alt key and hitting the letter A. And what I see on the scene is that that single cell has now formatted currency two decimal places.

Q. And now could you operate macro B?

A. Macro B, just to remind you, is the macro that had globally formatted the entire sheet as currency. So I'll hit Alt B to run that macro, and you'll see the same result as we had seen in 1-2-3. All the numbers in the table are formatted to currency, two decimal places. The last column does not display, and it shows as asterisks rather than as the previous values that had been there.

Q. Okay. Now, could you execute macro C, please.

A. I'll move my curse over to column G, which is the column I want to make wider to the width of 15. And I'll hit the macro Alt C, and now the column is widened.

You can see up again in the top left-hand corner that, you know, Quattro Pro 3 uses a similar notation to indicate that it's now with the 15.

Q. And now could you turn to the next page of your demo **[4-91]** file and look at the interactive macro.

A. So I'll press page down to move to the next set of macros, and I'll move my curse over again to cell C29, which would be in the top left corner of the table.

Q. Okay. Now, could you execute macro D for us, please.

A. Macro D just to remind you is the macro that chooses range format currency, and then prompts the user for the number of decimal places and the location.

So I'll hit Alt D. It prompts me for the number of decimal places as I did 1-2-3. I'll choose zero decimal places, and I'm

now prompted to the location I want to change. Quattro Pro 3 appears to use the term block, whereas in 1-2-3 we use the term range, but I'll highlight a certain number of cells. Again, that's three by three matrix. Now I'll press enter. And now there's nine cells are formatted as currency, no decimal place also.

Q. Thank you. Can you execute macro E, please.

A. Macro E as you recall is the macro that allows me to choose the type of formatting as well as the number of decimal places and the location.

To execute macro E, it's Alt E. It now pops up a menu box on the screen that provides what appears to me to be the same choices as I had in 1-2-3, so I can use my cursor key to move to the next choice. Interestingly, whereas I would have expected the up and down arrow keys to **[4-92]** work here, the right and left arrow keys also work as well.

So if I wanted to move from fixed, which is the first choice on the list, to scientific, I can press this right arrow key, which is what I'm doing now, and it moves what appears to be on the screen down one to scientific. Press enter to choose scientific. Again, I'll choose one decimal place, and I'll highlight the cells I want to change; and, again, I'll highlight those same four cells and now those four cells display a scientific notation.

Q. And now could you execute macro F.

A. As you recall, F is the one that did worksheet insert and paused to allow me to choose column and row. If I wanted macro F by hitting Alt F, I get a beep. I don't know if that was audible to anyone else in the courtroom, but the computer beeped and I get an error box that says menu unavailable, and I can just hit escape to recover from that error. The macro is done executing, but it didn't perform any operation.

Q. Okay. So that one didn't work?

A. That did not do the same thing as it had in 1-2-3.

Q. Could you try macro G, please.

A. To run macro G which was the macro that choose worksheet insert row and simply prompted me for the location of

the row, I choose Alt, and the letter G. It asks me where I choose to insert a row, and I'll just again default **[4-93]** as I did in 1-2-3 to the current location, and what it will do is right now there's a blank row between row 27 and row 29, where 28 is blank. If I press enter, it pushes the information down and now both row 28 and 29 are blank.

Q. Okay. Now what I'd like you to do next is switch the menus in Quattro Pro 3.0 to the 1-2-3 compatible mode. Can you do that, please?

A. Yes. To change the menus in Quattro Pro 3, I choose the slash key to access the menu. I'll move my cursor over to the top line menu command called options. I'll go down to the command called startup on that submenu, and select that by pressing enter. I'll move down to the option called menu tree and select enter. I then get a pop-up list that has three options in it: 1-2-3, Q1 and Quattro.

And you had asked me to selected 1-2-3?

Q. Yes.

A. So I'll press enter when the cursor is highlighting 1-2-3. And now what I see on the screen is that the top line menu has changed to be very similar to the 1-2-3 menu system.

Q. Okay. If I could ask you to back up—

MR. REBACK: Excuse me. Object to the witness's characterization as very similar. That screen speaks for itself, your Honor.

THE COURT: All right. I've seen it. The **[4-94]** objection is sustained. I'll strike that characterization.

Q. Okay. Could you back up one page to the A macro, please.

A. Page up to go back up to the top line.

Q. Now could you show us how these macros execute using the Quattro Pro 1-2-3 compatible user interface.

A. Okay. If I hit Alt A, the current cell is already formatted as currency. It may not be obvious what happens, but I can

hit Alt A and the macro runs. There's no error. There's nothing to indicate that it didn't work correctly.

Q. Okay. So without changing these back it's not visible. Can you quickly reset it or—

A. Yeah. I can just—I'm using the 1-2-3 menus in Quattro Pro to just reset this range.

Okay. Because I turned on global format, it's not a range setting, so I can go back in and choose slash worksheet—

MR. REBACK: Excuse me, your Honor. I object to this line of testimony. This interface is now out of the product. We're only here to try the Key Reader issue. He need not run through the Lotus compatible menu structure in order to try the Key Reader issue. If he wants to run the Key Reader, let's just let him do it.

THE COURT: Objection overruled.

A. Would you like me to change the global format back?

[4-95] Q. Yeah, if you can do it quickly.

A. I'm just going to do the same keystrokes I would have done in 1-2-3, which would be global format general which just puts that table of numbers back to the general display, which is no commas, no decimal places, no dollar signs.

Q. Now if you could show us in execution macro A, macro B and then macro C, please.

A. We'll run the macro, Alt A. It now formats that single cell, so C9 has turned to two decimal places. If I run the macro Alt B, it will now change the global format to currency, so the rest of the numbers on the spreadsheet will now display currency. So Alt B, and they display as well. Since I hadn't adjusted the column width, column G back to the prior previous setting, you know—

Q. But I take it you executed macro C it would change the column width to 15?

A. Absolutely.

Q. Could you flip to the next page, please, and just quickly show us the interactive macros?

A. Okay. Put my cursor now on C30, since I inserted a row to what was there previously. If I choose the Alt D macro, it prompts me for the number of decimal places, and I'll choose—in this case it was formatting the currency, so I'll choose one decimal place just to make something look a little different, and I'll highlight those nine cells **[4-96]** again.

Now you see I've formatted them as currency one decimal place.

Q. Could you try macro E, please.

A. If I run the Alt E macro, it gives me that menu I had described earlier of the different format options, and I'll choose scientific, again; and I'll choose zero decimal places this time and highlight four cells, and that is formatted as scientific.

Q. Could you try macro F, please.

A. If I run macro F holding down Alt F, it now pops up a little box that gives me a choice between column and row, and I can choose row and then specify the range, and I'll say the current range, and it inserts another blank row.

Q. And G.

A. If I hold down Alt G to run the G macro, this chooses worksheet insert row, and then pauses to allow me to specify the range. If I press enter, I get that range again.

Q. Now do I remember correctly that when you were using the Quattro native menus with the Key Reader, macro F was the one that did not operate properly?

A. That's correct. When I operated—when I executed macro F with the Key Reader turned on—excuse me, with the Key Reader turned off and the Quattro native menus, it did not execute correctly.

[4-97] Q. Okay. Now could you switch to Quattro Pro 4.01, please, or what we've been calling 4.01.

A. Again I'll use this Alt tab key combination in Windows is the appropriate product, and then select it.

We're now in Quattro Pro 4.01. I just had turned off the WYSIWYG display in order to set the number of rows and to

set the number of columns, and I'm in the native menus of this product.

Q. Okay. Could you turn the Key Reader on, please.

A. Yes. To do that, slash, to access the menu, choose tools, choose macro, move down to the Key Reader option, press enter, and then I'll select yes to turn that on.

Q. Okay. Now with the Key Reader turned on, could you try macro A.

A. Macro A, which is the format that the current cells turn to, two decimal places, I'll hit Alt A—press Alt A, and it formatted the current cell as currency two decimal places.

Q. Could you try macro B, please.

A. Macro B does the global format to currency, so all the values in the spreadsheet that aren't otherwise formatted will be formatted as currency two decimal places; so I'll hit Alt B, and you'll see the same result that we had seen earlier in 1-2-3.

Q. And could you use macro C to fix column G.

[4-98] A. Since column G is displaying asterisks, I will run Alt C and that makes the column a little bit different in the values in the spreadsheet.

Q. Now let's try the interactive macros on the next page.

A. So I'll page down and move to cell C29 again. The first macro, Alt D—excuse me or back slash D does range format currency. It prompts for the number of decimal places and the location.

To execute that I'll do Alt D. I'll choose zero decimal places, and I'll highlight those same nine cells I highlighted in previous products.

Q. I'm sorry. And now could you do E, please.

A. Macro E was the one that allowed me to choose the format type as well as the number of decimal places as well as the number of—excuse me, the number of cells or the location of cells. So I'll run that by choosing Alt E. As I did previously, point to scientific, choose scientific, choose one decimal place, and then highlight in this case only four cells. And those four cells are formatted as scientific notation, two decimal places—or excuse me, one decimal place.

Q. Now could you try macro F, please.

A. Macro F is the one that paused and prompted into either column or row, and I again do column and row. So to run that I choose Alt F, and I get an error message which **[4-99]** says menu unavailable.

Q. Could you please try the next macro of the one in which you have designated whether it's column or row.

A. I'll press escape to clear that error message, and I'm back to a ready prompt. If I choose Alt G to insert a row, it pauses and prompts me for where I'd like to insert the row. And I'll press enter, and it inserts a row at that location.

Q. Thank you. Now could we try the same thing in Quattro Pro for Windows, please.

A. Again I'll use Windows' hot key to move to Quattro Pro for Windows.

I had previously retrieved Quattro Pro for Windows—excuse me, actually Quattro Pro for Windows and had retrieved the demo DWK1 file from the floppy disk.

Q. Okay. Now could you just—Mr. Roshfeld, if you could just run through macros A through G, please, in Quattro Pro for Windows.

A. In Quattro Pro for Windows to run a back slash A macro I press the control key and the letter A. So I'll hold down control and the letter A, and it beeped and the worksheet seems to have disappeared.

Q. All right. Why don't we turn on the Key Reader.

A. (Complies.)

Let me open that file again from disk. And to **[4-100]** turn on the Key Reader I press the Alt key. To access the menu I'll move over to the option called property, select enter, move down to the option that says application, and select enter.

At this point I'm going to use the mouse because I'm just personally not comfortable with the keyboard conventions in Quattro Pro for Windows. I'm not familiar with them.

I'll choose the macro setting in this dialogue box, and then part of the dialogue box changes to give me the macro specific options; and I will click on the macro option called Key

Reader, and it puts a little check in the box next to Key Reader, and then I'll click on okay.

Q. Now with the Key Reader turned on, would you please execute macros A through G in Quattro Pro for Windows.

A. Just so it doesn't appear that I'm doing anything deceptive, I'm just going the make the size of this window like the spreadsheet number 45 and at the same time I'm going to pan it to partial black.

Now I'll execute macro back slash A by hitting control A, and now that cell is formatted as currency two decimal places.

Q. Okay. Now can you do—

A. Do you want me to do the C prompt?

Q. Yeah. Why don't you just walk through them, please.

[4-101] A. So the global format, macro back slash B is control B, and now the first four columns of values displayed as currency two decimal places, the last column displays as astericks.

If I put my cursor in that last column and choose the macro back slash C, by pressing control C, it now widens the column to a width of 15.

Q. And the interactive macros, please.

A. Page down to the interactive macros. And now the first thing I'll do is run the Alt D macro that allows me to interact and decide the number of decimal places and location. I'll choose control D. I'll choose zero decimal places, and press enter; and then I'll highlight those nine cells, three columns and three rows, and press enter. And now those nine cells are formatted as currency zero decimal places.

So to use macro E which allows me to choose the type of formatting as well as the number of decimal places and the location, I'll press control E, and I don't see anything—any sort of menu on the screen at all. On the bottom right-hand corner of the screen, though, I see two indications; one says macro, which I believe indicates that I'm still in the midst of the macro, and next to it is an indicator that says menu. So perhaps I'm still on the menu.

So what I'll do is blindly execute the same keys **[4-102]** that I would have done previously. So I'll choose S for scientific, and now it asks me for the number of decimal places. I'll choose one, and press enter, and then highlight the four cells to be modified and press enter. And now those four cells appear to be formatted the same way as they had been previously, scientific notation, one decimal point.

Q. So am I right in understanding that that interactive macro accepted the same keystrokes at that point as 1-2-3 would have?

A. Yes, it did.

Q. And produced the same result?

A. Yes, it did produce the same result.

Q. Now could you try macro F, please.

A. At this point I'll run macro F, which allows me to choose between column and row insertion in 1-2-3. I'll hit control F. I don't see anything on the screen, but it tells me that it's in a macro and it's ready. So I'll choose R for row—actually I may have hit the wrong key. I'm going to hit control break to stop the execution of that macro, because I'm not absolutely sure I may have hit control. I may have hit another key.

So I'll hit control F, and I'll be more careful this time. Now on the bottom of the screen in the far right lower corner I see macro and menu again. And what I would **[4-103]** do in 1-2-3, if I were doing this I would press R for row, and now I get a prompt that says enter row, insert block. So I'll accept the current location, and it appears to have inserted a row exactly where I wanted it to and operated in the same way as if I had done that sequence in 1-2-3.

Q. And am I right that this is the macro that didn't work in Quattro Pro 3.0 and 4.01 in the native mode. Is that right?

A. That's correct.

Q. But it works in Quattro Pro for Windows?

A. Yes, it appears to work correctly in Quattro Pro for Windows.

Q. Now could you try macro G, please.

A. Macro G just inserts a row, but it allows me to choose where. So I'll choose control G. And it just popped up with that same prompt for enter row, insert block, and I'll press the current location, and now it inserts another blank row in there.

Q. So am I right that each of these interactive macros worked in Quattro Pro for Windows with the Key Reader on?

A. That's correct. Each of the macros worked correctly in Quattro Pro for Windows with the Key Reader on.

Q. Okay. Now each of the macros you've created to this point have been written by means of the first letters of the commands. There's another way to create macros, isn't **[4-104]** there?

A. That's correct. Another way to create macros is actually use cursor movements to walk through the menus.

Q. Okay. Let's return to Lotus 1-2-3, and if you could show us some macros that were created using cursor movements rather than the initials of commands.

A. Okay. Again, I use the Windows hot key to go to 1-2-3, and I'll press page down to go to that third page or third screen of the macro information.

You see a similar table of data that we had had been working with previously, and you can see that there's two macros, back slash H macro and a back slash I macro. Let me describe the back slash H macro.

The first command in that macro is a slash which brings up the menu; but at that point rather than doing first letter selection, the way some users would work through the menu, it actually moves the highlight through the menu similar to the way I was doing it earlier to illustrate some of my points.

So if I were manually doing this macro, I would put my cursor on the first cell containing data. The slash key brings up the menu, the command that you see in macro back slash H right would move the cursor right one location. The tilde performs the same operation as if I hit enter. There's then a second tilde, which is equivalent to a second **[4-105]** enter,

which accepts the current menu choice which is format. Then there's two commands right, and followed by right again, so that moves the cursor right two positions to highlight the menu choice currency. And then that's followed by a tilde to be the equivalent of enter, and then at that point the macro includes the number zero, so it would replace the two that's the default setting there with a zero. And then you would press enter once to format that—excuse me, enter to accept the zero, and then enter a second time to accept that range.

And I'm going to—rather than finish the execution of this command sequence manually, I'm going to use the escape key to escape back out of the menu, and now I'll run this Alt H macro.

Q. Okay. Would you do that, please.

A. I'll hold down the Alt key and hit the letter H, and now the format of that single cell changed to display rather than two decimal places in currency it displays as zero decimal place in currency.

Q. Okay. Now, is there something in 1-2-3 that allows you to slow down the operation of the macro, the execution, so you can see it executing step by step?

A. There's a feature in 1-2-3 that we call trace mode. The trace command allows somebody who's creating a macro to walk through the macro as it executes one step at a time **[4-106]** which is useful for finding any problems or mistakes in writing the macro.

Q. Could you turn on that step mode, please, and then could you execute macro I using the step mode so that we can see how it works.

A. To turn on step mode, I hit Alt and the function key F2. And you'll see on the bottom of the screen the word step appears on the lowest line. That just indicates that I've turned on the step mode.

And now which macro would you like me to execute in the step mode?

Q. Let's do I.

A. The I macro changes the column width of the current column to nine. So what I'll do is I'll run this off the I macro. That puts on macro, but because step mode is on you'll be able to see it execute one instruction at a time.

So if I hit Alt I, the indicator on the bottom of the screen turns into a blinking SST, which indicates that I'm in single step mode now. I press the space bar. It executes the first command and in the macro which indicates slash which brought up the menu.

If I press space bar a second time—what we're looking at is the macro in cell B45, the back slash I macro. The first command, which is slash; the second command is tilde, equivalent of enter; so when I press space bar, which **[4-107]** is the equivalent of enter, which brings us to the worksheet submenu.

Now if I press the space bar again, it will execute the next command in macro, which is a right. So it moves the menu cursor one position to the right. I press space bar again. There's another right, and then a third right. So I've moved cursor three positions to the right to highlight the word column.

At this point, the next step in the macro is a tilde, which is a return key suppressing the space bar, allows the macro to execute that tilde which does return to choose the column menu. It then does another tilde or another return to accept set width. It's then prompting the user for the column width; so when I press a space bar, enter the number nine from the macro, and then finally accept that, so when I press the space bar one more time, see the width of column G, which is where the cell pointer is currently will change to nine.

So the way that macro executed was by moving through the menu tree, but rather than moving through the menu tree, using first letter selection it moved through the menu tree by moving the cursor to highlight the option that the user was interested in.

Q. Now I'd like to move to Quattro Pro 4.01.

A. All right. So I will again use the Windows hot key **[4-108]** and go to Quattro Pro 4.01, and then I'll page down to get to that same part of that demo worksheet.

Q. Now, is there something in Quattro Pro that is comparable to the step mode in 1-2-3 that allows you to see a macro in operation step by step?

A. There's a macro debugger facility that provides a similar capability to what we call step mode in 1-2-3.

Q. Okay. Actually before you turn that on, could you just trace for us—if you were to do on the Quattro Pro 4.01 menu that you see before you the cursor movements indicated by macro H, could you show us the level of blank, please.

A. I'm going to walk through manually the same keystrokes which are represented in cell W42, which is the back slash H macro key.

So the first thing is a slash key which brings up the menu. And as you recall, this was the macro that changed formatting to currency zero decimal places of a single cell. Slash would bring me up to the menu, highlighting the word file. Right would highlight the word edit, return by the tilde, I choose the edit menu—if you'll excuse me for a moment. I need to refer to my printout here because I can't see the macro any longer. So return would accept the edit menu.

MR. REBACK: Excuse me, sir. You referred to **[4-109]** a printout. What was that, Sir?

THE WITNESS: What I have in front of me is an outline to sort of remind me of the steps we're going through.

MR. GUTMAN: Mr. Reback is welcome to a copy, but I'd really appreciate it if he didn't interrupt the demonstration. We're trying very hard to get done before the lunch break.

MR. REBACK: Might I just see it for a second?

THE COURT: You may.

MR. REBACK: I just want to know what it is.

THE COURT: You may.

MR. REBACK: I'm sorry. Where were you talking about, sir?

THE WITNESS: I think on page 8—7. Excuse me. Because I'm trying to recreate this macro manually, and by doing so I brought something up on the screen that covers the macro itself. I'm referring to the piece of paper to remind me what the macro was.

MR. REBACK: Thank you, Sir.

A. So the macro was slash to bring up the menu, right, followed by two tilde, two enters, so the first chose the edit menu, the second will choose I suppose copy. That was followed by the command right two times, so I would choose right, right, and it appears that I'm now in the copy menu **[4-110]** in the Quattro Pro menus, and then choose enter, and at this point it prompts me to type in the macro, we'd type in a zero if we're in 1-2-3 for zero decimal places. It doesn't make sense. We're not in the right context here, and then I would hit enter and get an error message because it's prompting me for cell location and I'm typing in a value.

Q. So the cursor movements described in the macro would not work on the Quattro Pro native menus. Is that right?

A. It appears to be correct, yes. If I walk through them it appears to be totally different demands.

Q. Now, could you turn on the Quattro Pro equivalent of step mode, please, and show us how macros H and I execute.

A. If I hit the slash key, move over to the tools menu, choose macro from the tools menu, move down to the choice for debugger, choose the debugger menu, and say yes, now on the bottom right of the scene it says debug which indicates that I've turned that feature on in Quattro Pro, and now you'd like me to execute—

Q. Yes. Could you first execute macro H and then macro I, but—

A. To execute macro—

Q. I'm sorry. Have you turned on the Key Reader? I don't remember.

A. Yes, I believe we have. Let me just confirm that. Slash tools, macro—yes, the Key Reader is turned on.

[4-111] Q. Fine. Now could you try macros H and I to see whether they execute.

A. H would in 1-2-3 format, the current cell as currency zero decimal places. So Alt H pops up what they call a debug window and it says trace window on the bottom, and then I can press the space bar to walk through this macro.

The first few times I press this space bar nothing appears to change on the scene. Now when I press the space bar I get that format menu that we had seen earlier in some of the macros, some of the interactive macros.

Continue to move through this and what happened as the macro said right, it moved right through the menu.

Q. Which in this case is down?

A. Which in this case is down, that's correct. And then it prompts for the number of decimal places and accepts that. So the net result of this macro is the same as it has been in 1-2-3.

Q. Now could you try the same thing with macro I, please.

A. Yeah. Macro I was the macro data for the column width, so I'll go over to column G that has that wired column and I'll hit Alt I to run the macro. I hit that debug window, trace window. I'll press the space bar to walk through that, and so what's happening in that debug window is that each time I hit the space bar a different **[4-112]** part of the macro, a different individual command in the macro is highlighted, and I'll point out if I see something happen on the screen other than that.

So it's moving through the macro in the debug window. It's not apparent to anything that's happening. Now when I get to the point where I'm highlighting the number nine, which in 1-2-3 is where I would enter, the column width to be nine characters, I see a prompt on the top of the screen asking me to alter the width of the current column. The nine goes into that entry, and then it accepts it so the end result is that the macro generated those same results in Quattro Pro.

Q. Okay. Now let's try the same thing with Quattro Pro for Windows, please, macros H and I.

A. So macro H—

Q. And again before we do it, could you please check to make sure that the Key Reader is turned on; and if Quattro Pro for Windows has something comparable to the step or the debugger, could you turn that on, please.

A. First I'll check for the Key Reader, so I'll go into their property menu, choose application. Again I'll use the mouse to choose the macro settings, and I can see that the Key Reader is in fact turned on.

Q. So it stayed on from the last time?

A. Yes.

[4-113] Q. And now could you—

A. Now I'll go into the menu again and choose tools, macro, and I'll turn on their debugger.

Q. Okay. Now would you please step or debug your way through macros H and I.

A. Pull down the control key and press the letter H—

MR. REBACK: Excuse me, your Honor. I do object to that question. Stepping and debugging are not the same thing. Lotus can only step. This product can debug.

THE COURT: The objection is overruled. I understand the distinction you're making. I've seen it, but the objection is overruled.

MR. REBACK: Very well, Sir.

A. I'll press down the control key and hit the letter H. I get some sort of dialogue box coming up with a number of commands in it, as it did in Quattro Pro for DOS it highlights the individual commands. If I press the space bar to do the equivalent, I get to the slash menu in 1-2-3 it executes the slash, and I see this box pop up on the screen. It's vertically longer than it is horizontally wide. There appears to be some sort of dark color at the top of that, and then some other cursor of some type or other dark colored box within that box, but I don't know what that is.

If I press the enter key again to generate a **[4-114]** right in the 1-2-3 menus, this would have moved across the menu. If

I press the space bar that cursor that I saw in that box that popped up moved down what looks to be one row or so but it moved down a position.

If I press the space bar again to execute the next command in the macro, which was a tilde or a return, the box appears to have changed shape—well, not so much shape, but just seems to be higher or longer vertically than what it was before; and, again, the cursor seems to be at the very top of that box.

If I press the space bar again, which was another tilde in 1-2-3 which would have chosen another menu in 1-2-3, this time the box changes both height and width and the cursor itself changes size in the box, and again the cursor is inside the box.

The next two commands in the macro were right, right, which in 1-2-3 moved me in the format menu two positions over to highlight currency. If I press the space bar, what I see is that that cursor was in that box that's popped up on the screen has moved down what appears to be one position; and if I press the space bar again to execute the second right command, it moves down another position.

At this point if I continue to press the space bar one more time I get a dialogue box of some sort that prompts me for enter number of decimal places. If I press **[4-115]** space, again, the macro enters a zero into that dialogue box, and then pressing the space bar one more time accepts that zero; and finally the last column—the last command in the macro accepts the current cell as the cell to be modified. So the end result in this was that I got to change single cell formatted as currency zero decimal places as I had in 1-2-3.

Q. Okay. Now could you do the same thing please stepping through macro I.

A. I'll move my cursor over to column G, which is the column that was—or is currently 15 characters wide. I'll choose control I. Again, I get the macro debugger box on the screen in Quattro Pro for Windows. Pressing the space bar which is the first command in the macro pops up, again one of these boxes with no words in them, but it's long, long vertically, narrow horizontally and it appears to have some sort of cursor in it.

Pressing the space bar again does an enter command and it's not obvious to me that anything has changed in that box at this point. So space bar again does a right command in the macro, and that cursor in that box seems to move down one. Space bar again makes it appear to move down another position. Third time makes it appear to move again which is—1-2-3 will go those three right commands we've done. It's moved me horizontally through the menu to choose **[4-116]** a particular submenu.

The tilde in 1-2-3 would have chosen a submenu, and now I see that box has changed size again with the cursor at the top of it, which has only four or five lines in it, defining the line and inserting the height of that cursor. Enter again brings up a dialogue box that prompts me to alter the width of the current column. The macro then puts the value nine in there and accepts it, and the end result is the column is now nine characters wide.

Q. Mr. Roshfeld, the macros you've been showing us so far have been fairly simple. Have you prepared one that is more complex?

A. Yes, I have.

Q. Could you switch back to Lotus 1-2-3, please, and show us that macro.

A. I use the Windows hot key to move back to 2.01. I page down, yet, again, you can see on the top part of this screen a much more complex macro back slash J.

Q. And before I ask you to execute it in step mode, let me just ask a couple of questions about some of the notations here.

In curly braces on the first line it says: Go to start. Could you explain that, please?

A. The reason that's in the macro is that the cursor goes to the appropriate position at the beginning of the macro. **[4-117]** So the go to command within braces is something that's built into 1-2-3. It's just a built-in command that allows the user to move the cursor in the spreadsheet; and I've defined a range name called start to just go to a particular location.

The reason I put this in the macro is because I was inserting in rows all over the place. I wanted to make sure when I started the macro it went to the correct location.

Q. Okay. And on the next line of the macro, second set of curly braces in, it says right two. What's that notation mean?

A. 1-2-3 allows you to do this rather than having to put the word right multiple times. So if I wanted to do six rights, rather than having to say right within braces six times, there's various short cut notations. One is to follow the word right by space and number in getting the number of positions.

Q. So right two or right eight means move the cursor wherever—

A. Move the cursor wherever it is whether in a menu or on the worksheet right that many positions.

Q. Okay. I see end in the same line. What's that?

A. End is a command in 1-2-3 that can be used either manually or in a macro in conjunction with another cursor **[4-118]** movement command to the move to the end in a particular location.

So, for example, if I had a long column of numbers and I wanted to delete that whole long column, in the range erase command when this prompted me for location, I could choose end down to indicate that the macro should highlight all the way down to the bottom of the column.

If I use end in combination with right, it would highlight all the way to the right, and up and left would work the same way in conjunction with the end command.

Q. Now what does this macro do before we see it.

A. This macro will do essentially three things. The first is it will move the cursor to a location called start, which is one of the options in the tables that you see in here.

The second thing it will do is change the formatting of the numbers in that table from what it's currently global formatted or currently two decimal places to a format of currency zero decimal places. It will then reset—it will then go into a graph menu and do a reset to clear out any previous settings that had been there, sort of house keeping; and then it will build a graph from scratch after its reset all the settings

by choosing the type of graph setting, the individual ranges for X, A, B, C and D. The X range would be the product names. A, B, C and D will **[4-119]** correspond to those quarters shown in that table, and then once it builds the graph it will then display the graph on the scene.

Q. Okay. Now could you—and in the interest of time you could just step through it without describing each step just slowing it down enough so that we can all hear. If you'd step through the execution of macro J.

A. Alt G to Alt J, and now I'm going to step through it. And what's happening is the macro is executing, and those commands such as right and left are moving the cursor through the menus.

One thing I'd like to point out is that the macro system is intelligent enough that if it says right space four or right space two, it will not refresh the scene in the midst of that cursor movement, which is why it appears when I'm in step mode that the macros are jumping around a little bit, because internally it's doing that movement but not redisplaying each time.

So I've chosen the type of graph. Now I'm choosing the X range, choosing the A range, which is key one; the B range which was key two; the C range which was key 3, the D range, which was key four, and now viewing the graph.

Q. Okay. Now, could we quickly move to Quattro Pro Quattro 4.01.

[4-120] A. Yes. Okay, in Quattro Pro 4.01 I have the exact same file, so it's the same macro.

Q. Could you check to make sure Key Reader is on and the debugger is on.

A. Slash, tools, macro.

Q. It stayed on?

A. Yes. The debugger is still turned on and the Key Reader is still turned on.

Q. Great. Now if you could again quickly step through this macro in execution in Quattro Pro 4.01.

A. So I'll choose Alt J to execute the macro, and I'll somewhat briskly press the space bar to move through the macro.

You see at this point that it did pop up a menu and prompted me to this number of decimal places to format, but for many of these other commands I don't necessarily see a menu coming up. I did just see a menu for the type of graph, and I am seeing some changes to the prompt line indicating which graph series that I seem to be setting, and the last result is that it displays that graph.

Q. Okay. Could we try the same thing in Quattro Pro for Windows, please.

A. Okay. Press page down. I assume you want me to check on the Key Reader?

Q. Please.

A. Lucky guess.

[4-121] Alt key to access the macro—the menu, choose tools, I'll choose macro. This option allows me to disable the debugger which tells me the debugger is already turned on, so I'll escape out of that. Actually on the bottom right of the screen you can see debug. I can also choose the property application as I did before, choose macro with my mouse, and I can see that the Key Reader has stayed on.

Q. So that stayed on?

A. Correct.

Q. Okay. Now if you could step through the execution of this macro, please.

A. On this macro, back slash J macro, Quattro Pro for Windows is control J. I see the macro debugger window.

First thing it does is go to the range called start. And now as it's executing commands that in 1-2-3 would have walked it through the menu, I've seen some of these boxes come up on the screen and the cursor appears to move. Sometimes the box changes size and shape, sometimes the cursor appears to move within a particular box.

Occasionly I see some sort of indication at the top of the screen that it's changing a particular graph series by popping up a little—I don't know what they're calling it in Quattro Pro. It looks like a little dialogue box or input box; and again the menus seem to come back as I walk through the macro.

[4-122] And the last step of the macro is to do that same graph. The graphs do appear slightly different in each of the products, but they're all stack bar graphs showing the same data with the same X range and the same data points.

Q. Okay. Now, the last two macros you prepared, if you can escape out of the graph. Let's move to your last two macros, macros R and W, and maybe we can learn something about those boxes that appear on the screen for Quattro Pro for Windows.

A. The macro—

Q. Could you tell me what macro R does to it?

A. The macro back slash R does the slash which in 1-2-3 would bring up the menu. R, which chooses the range command from the top line menu, and then pauses using that curly brace with a question mark inside for user input; so if I ran this macro in 1-2-3, I would get slash range, and then I could do whatever I wanted from inside the range menu.

Q. Let's run it in Quattro Pro for Windows and see what happens.

A. So to run that macro I choose control R. I turn off the debugger or—

Q. Yeah, why don't we turn off the debugger so that we can see that.

A. Let me break out of this, go to the tools menu again, choose macro and choose to disable the debugger.

[4-123] Q. Now let's try macro R, again.

A. If I choose macro R by pressing control R. The only indication that something's going on is on the bottom right corner of the screen it says macro and it says menu. At this point if I were in 1-2-3 I would know that I could choose F for format, S for scientific, put in one decimal place, and enter, and then highlight a range to format.

Q. And it worked?

A. Yes, it did. I was able to do the same keystrokes from that point on that I would have been able to do in 1-2-3.

Q. Let's try macro W and see what happens when you use that.

A. Okay. Macro W simply does a slash command to bring up the menu, and the letter W to choose the worksheet menu command; and then it pauses once I'm inside the worksheet menu.

So if I run that macro by choosing control W, again I see on the bottom right-hand corner of the screen it says I'm still in the macro and the menu, and I could choose let's say I for insert, R for row, highlight where I wanted to insert a row, perhaps underneath those titles there, and press enter and it would insert a row in that location.

Q. Now, can you move back and forth using macro R and macro W? Can you move back and forth through the levels of **[4-124]** the menus as if you were in 1-2-3?

A. I believe I can. If I choose to run that same back slash W by pressing control W, and I choose insert, I now see a box that looks like it has two choices in it. In the 1-2-3, that position would be column and row. If I escape back out at another location I could choose global default, other, international, or I could escape back out and choose column, set width, I may alter the width of a column. So, yeah, it appears that I can, and I just broke out of that macro at the same time. I can work my way through those menus using the same keystrokes as I could in 1-2-3.

MR. GUTMAN: I have no further questions, your Honor. Thank you, Mr. Roshfeld.

MR. REBACK: We are going to have some questions. Might we take the break now sir?

THE COURT: All right.

MR. REBACK: May we have the break till 2:30? Would that be too difficult?

THE COURT: No, I'll permit it.

MR. REBACK: Thank you, sir.

THE COURT: All right.

MR. GUTMAN: Thank you, your Honor.

THE CLERK: All rise.

(Lunch recess taken at 12:44 p.m.)

* * *

[4-126]

DIRECT EXAMINATION

BY MR. REBACK:

Q. Sir, could you once again state your name for the record?

A. Yes, Robert Warfield.

Q. And could you briefly describe your background for the Court.

A. Yes, I grew up in Midland, Texas, and went on to get my bachelor's in computer science at Rice University.

Q. Where do you work now, sir?

A. I work at Borland International.

Q. What is your position there?

A. Vice-president of research and development.

Q. Are you familiar with the product known as Quattro Pro for DOS, sir?

A. Yes, I am.

Q. And did you have a role, any role in the development of that product?

A. Yes, I was responsible for that product.

Q. Sir, would you please give us a demonstration of the most current version of that product?

A. Sure. That would be Quattro Pro DOS Version 4.01. [4-127] Let me just bring it up here.

(Pause.)

Okay. As you can see here, we have the program on the screen.

Q. Yes, sir. Is that in the default mode?

A. Yes, this is the way the program normally comes up as installed.

Q. Okay.

MR. REBACK: I want to do just a couple of preliminary things just to establish the record here, your Honor.

THE COURT: All right.

Q. Can you show us where the menus in Quattro Pro for DOS are found, sir?

A. Yes, the menus are along the top of the screen here where I'm pointing with the mouse cursor.

Q. Are those two-line moving cursor menus?

A. No, these are pull-down cascading menus.

Q. How does the user select a menu, sir?

A. Well, for example, if I wanted to bring up the graph-type menu, I would place my mouse cursor over graph, click the left button, move down within the pull-down menu to graph type, click on graph type, and that would then bring up the cascading graph-type menu.

Q. All right. Sir, can you take us up to the top line of [4-128] the menu bar, please.

A. Certainly.

Q. These menus that I see across the top and on the screen, are they 1-2-3 menus?

A. No, this is our native menu tree.

Q. Can you demonstrate to me the 1-2-3 menus?

A. No, I cannot.

Q. Why can't you?

A. The 1-2-3 compatible menus are not included with Quattro Pro 4.01.

Q. Are the 1-2-3 menus anywhere in the current version of Quattro Pro for DOS?

A. No, they're not.

Q. Okay. Now, for the purpose of demonstrating a couple of things to the Court, if you would please move the cursor down to the line beneath the menu bar.

A. Okay.

Q. Could you tell us what that is, sir?

A. This line here that you see across the screen is something we call the speed bar. It contains a series of buttons that the user can click on with a mouse in order to initiate commands.

Q. Is that part of the menu, sir?

A. No, that's part of the Quattro Pro user interface.

Q. How does the button bar work again?

[4-129] A. the user would simply bring their mouse cursor over the button they're interested in and click the left mouse button.

Q. Can you use these buttons without a mouse?

A. No, you cannot.

Q. Are you familiar, sir, with Lotus Versions 1, 1A or 2.0?

A. Somewhat.

Q. Are you familiar with Lotus Version 2.1 that you saw today?

A. Yes.

Q. Does that work with the mouse?

A. No.

Q. Does it have buttons, to your knowledge?

A. No, it does not.

Q. You know, when Mr. Roshfeld was demonstrating this product, I believe the product now before us, I believe he said that it was in text mode. Is that the default mode for the product?

A. Not for this machine, no.

Q. Could you tell us the steps you would go through and demonstrate them to put it in text mode.

A. Sure. In order to do that, you bring your mouse cursor over to the options menu, click with the left mouse button, move down with the cursor to where it says display [4-130] mode and click once again, and here we see all the different modes listed. In order to go to text mode, we would select in mode 80-by-25 by placing the mouse cursor over it and again clicking the left button, and now you see Quattro Pro displayed in text mode.

Q. And this is the way Mr. Roshfeld demonstrated the product?

A. Uh-huh.

Q. You have to be verbal, sir, for the record.

A. Yes, sir.

Q. Thank you.

Could you take us now back to the default mode.

A. Certainly. I do that in just the same way. Go back to the options menu with the mouse cursor, click, move down

to display mode. Within that cascading submenu I move down to WYSIWYG and I click, and we're back now in graphics mode.

Q. Sir, was Quattro Pro optimized to be used with a mouse or keyboard?

A. Quattro Pro was optimized to be used with a mouse.

Q. Bearing in mind the Court's admonition to me earlier, what does optimized mean?

A. That just means that it's a lot easier to use the product if you have a mouse.

Q. Thank you.

Can you run a Lotus macro with this version of [4-131] Quattro Pro DOS?

A. Some macros will run if you turn on the Key Reader, yes.

Q. That's the feature that Mr. Roshfeld demonstrated earlier today?

A. Yes.

Q. Where in the native mode menu tree would I find the Key Reader, sir?

A. Would you like me just to turn it on?

Q. Yes, please go ahead.

A. I move up to the tools menu with the mouse and click again, move down to the macro submenu, click, and down at the bottom of this menu we find Key Reader. If we click on that, it says no/yes. To turn it on we click on yes, and now the Key Reader has been enabled.

Q. Who created the Key Reader feature?

A. I came up with the idea for the Key Reader after I was asked by Philippe Kahn and Bob Kohn to try to find some way to run Lotus-compatible macros under the native menu trees that were similar to Excel's translator feature.

MR. GUTMAN: Your Honor, excuse me. We had a stipulation that the only live testimony was going to be demonstrations. They called Mr. Warfield as their person to do the demonstration. Instead what he is trying—what the questions elicit—what the last question clearly called [4-132] for is his testimony concerning the history, and I thought that our

stipulation provided that that was not what we were doing by live testimony and that the record on these matters was closed.

MR. REBACK: The record hasn't even been offered yet.

THE COURT: Well, wait a minute. Under the stipulation, wasn't it agreed that the record would not include live testimony?

MR. REBACK: No, sir, we were entitled to one live witness for a demonstration.

THE COURT: Oh, but what the objection is is that you're trying to use it not for a demonstration but for testimony that is not explaining the demonstration, but is giving other evidence bearing on other issues in the case than just the demonstration.

MR. REBACK: I see.

MR. GUTMAN: And I would—

MR. REBACK: I don't know what to say on that point. I thought I was entitled to do that. That's the reason I have this witness.

THE COURT: Let me see the stipulation.

MR. REBACK: My point about this, it provides context for what's shown on the screen, but we'll hand up to you a copy of the stipulation.

[4-133] (Document handed to the Court.)

MR. REBACK: The relevant paragraph I believe is paragraph 4, your Honor.

MR. GUTMAN: Also paragraph 3, your Honor, in which both sides waive any live testimony apart from the demonstration.

THE COURT: Well, I think I must sustain the objection. As I read the stipulation, the purpose of the witness to conduct the live demonstration is to conduct the live demonstration, not to talk about other issues in the case.

MR. REBACK: All right, your Honor. I apologize and I will keep that ruling in mind as I move forward.

MR. GUTMAN: I would move to strike the last answer, if I may, because it was clearly contrary to the stipulation.

THE COURT: Motion sustained.

MR. GUTMAN: Thank you.

BY MR. REBACK:

Q. Mr. Warfield, what does the Key Reader do?

A. Well, quite simply, the Key Reader takes the key strokes embodied in a macro and using its own internal data on the overall hierarchy and the pickletters involved there, it then translates those keystrokes into our menu **[4-134]** equivalents and turns those menu equivalents over to the program for execution.

Q. What version of 1-2-3 macros was the Key Reader intended to run, sir?

A. Version 2.01.

Q. Will it run Version 2.0 macros?

A. Don't know.

MR. GUTMAN: Your Honor, apart from the fact that the computer is turned on and that Mr. Warfield is sitting with a keyboard in front of him, it would be hard to tell this was a demonstration.

MR. REBACK: I'm talking about features of the product now, your Honor. I mean, I really am. What macros does it run, what macros doesn't it run, how does it run them.

THE COURT: Well, I don't think that's correct. Mr. Reback, I think you're fighting your own stipulation.

MR. REBACK: Well—

THE COURT: I want an explanation of what's going on here, not an explanation of what's not going on. That's not a part of the demonstration unless it helps me understand the limitations of the demonstration itself, and the issue about 2.0 and 2.1 is not within that category.

MR. REBACK: If I had phrased the question to **[4-135]** ask the witness if he could run a 2.0 macro, would that meet your Honor's objection?

THE COURT: I don't see how that would make a difference. That's not asking me about what—something about this demonstration. It's asking me about another question on which we've had some further colloquy earlier today. And I understand that you would like to show me that features of

this demonstration do not apply to 2.0, but that's not a part of the demonstration. If you want to offer testimony on that subject, let's face up to it that that's what you're trying to do.

MR. REBACK: I will separately do that, then, your Honor.

THE COURT: All right.

MR. REBACK: I apologize.

BY MR. REBACK:

Q. Mr. Warfield, what does the user—strike that, sir.

This morning we saw Mr. Roshfeld run a number of macros that he had created for the demonstration. What does the user usually or typically see when the Key Reader is running a macro in this product?

A. It just runs the macro and you would see the native Quattro Pro user interface and whatever results the macro produced.

Q. Is there a feature in this product that we can **[4-136]** demonstrate to show the Court how the Key Reader works?

A. Sure. I think our menu editor command would be appropriate for that.

Q. Please do that, sir.

A. Okay.

Q. Please demonstrate that feature for the Court.

A. So I'll move over to the options menu with my mouse and I'll click, and then I move down to startup and click again. Near the bottom of this menu we see the edit menus command which I'll click on, and what this causes the program to do is for its menu editor to take the menus that were currently in use in the product and bring them in and display them so you can see how they look.

Q. All right. Now, I see some menus on the screen now on the spreadsheet portion of the screen. Are those Lotus menus?

A. No, these are native menus.

Q. Where are these menus stored, sir?

A. These menus are stored in one section of the QUATTRO.MU file.

Q. Now, are the words that we see there part of the Key Reader portion of that file?

A. No.

MR. GUTMAN: Your Honor, again, now we're talking about the internals.

[4-137] MR. REBACK: It's demonstrated in the product, your Honor, as shipped.

THE COURT: Let me hear the objection first and then I'll hear your response.

MR. GUTMAN: Your Honor, Mr. Warfield is now being asked to describe the internal file structure of the program. Again, there is an effort to supplement the record that we had stipulated was not being supplemented by live testimony. There are documents which I understand they want to offer on this subject, there's deposition testimony of this gentleman that's been taken, but this is not a demonstration when he asks now what file inside the program is that—

THE COURT: The objection is overruled.

Q. Where are the Key Reader menus—I'm sorry. Strike that.

Where is the Key Reader data displayed in this menu editor, sir?

A. Well, what you'd have to do to see the Key Reader data—and I'll just do it here as I'm talking—is move the cursor down past all these words that are part of the native menus until you get to this bar here. The bar has a plus on it signifying that if I strike the plus key, I can then open up the bar and cursor down into this section below, and now we're in the Key Reader section.

[4-138] Q. I don't see any menus there, sir.

A. Well, there aren't really any menus here. The Key Reader just contains an embodiment of the hierarchy that's needed there and you can see in this panel at the top that I'm pointing to with the mouse that there's also a menu key indicated, a portion called actions which is basically what we call our menu equivalents, and a few other pieces of information.

Q. Okay. I see something to the right called menu key at the top there. Could you move the cursor over to that?

A. Certainly. That's right here.

Q. I've heard the term pickletter. Is that the—

A. Same thing, yes.

Q. What is the relationship between that W where the cursor is and that highlighted red plus sign on the screen?

A. Well, if you will, this W is the letter that has to be encountered at this point in order to proceed to the next level during macro execution.

Q. What does the W stand for?

A. For worksheet.

Q. For the word worksheet. From the 1-2-3 menu command hierarchy, sir?

A. Correct.

Q. Is the word worksheet anywhere in the Key Reader portion of the QUATTRO PRO.MU file?

[4-139] A. No, it is not.

Q. Now, using the menu editor feature, please demonstrate the order of the pickletters as they appear in the Key Reader portion of the QUATTRO.MU file.

A. Do you want me to do that specific macro?

Q. Just open it up so the Court can see how it looks opened up.

A. So here's the first level here on the left. We're sitting on W. If we hit plus, that opens up and you can see we go on to the next levels.

Q. Okay. Could you go up to the first plus on that column.

A. Okay.

Q. What pickletter is that plus, sir?

A. That one is a G.

Q. Can you show me where on the screen that's demonstrated?

A. You can see that if you look again up at the top right next to where it says menu key there. I think it's in red. You can see a G.

Q. It is in red. Thank you.

Now cursor down to the next plus.

A. Okay.

Q. What pickletter is that?

A. Now we have the pickletter I.

[4-140] Q. Okay. Could you cursor down to the next two and tell me what they are, sir.

A. We then have a D followed by a C.

Q. Could you open up the C, sir.

A. Sure. If I hit the plus, C opens up.

Q. I don't see anything else there. Are there any other pickletters?

A. There are some other letters you can see by moving the cursor to the right. There are no pluses displayed because there's no further submenus, but you can see as I move around here, there's an S, there's an R, there's an H. There's a bunch of them here.

Q. Is the order in which these pickletters are listed here in the Key Reader portion of the QUATTRO.MU file the same order as the first letters appear in the Lotus 1-2-3 menu screens?

A. No.

Q. All right. Is this something that you wrote for this demonstration, sir?

A. No, this is a facility that's available to all Quattro and Quattro Pro users.

Q. If we went across the street to a software store and purchased a Quattro Pro current version, this would come with it?

A. It would be right there.

[4-141] Q. Thank you.

I believe you just told me that the order of the pickletters is not the same as the first letters in 1-2-3. Does it reflect the order of the 1-2-3 commands in any respect?

MR. GUTMAN: Objection, your Honor. I mean, I believe he's—

THE COURT: I'll permit that question.

A. Only to the extent that that order is relevant for macro execution.

THE COURT: Well, I think I better not permit the question in light of the answer that obviously is going beyond the demonstration to an expert opinion. And the stipulation I read

a moment ago said we were not going to have an expert as the demonstrator, didn't it?

MR. REBACK: Yes, sir.

MR. GUTMAN: Yes, your Honor.

THE COURT: I don't want any expert opinions, then, about the case.

MR. REBACK: For purposes of guidance, could I just hear the last question and answer to which the Court sustained an objection?

THE COURT: You may.

MR. REBACK: Very well, your honor.

(Last question and answer read back by the court reporter.) **[4-142]**

Q. Suppose that—strike that.

Earlier today I saw Mr. Roshfeld demonstrate a macro that he had written with the keystrokes RFC. Do you recall seeing that, sir?

A. Yes, I do.

Q. Could you demonstrate how the Key Reader would just translate the letter C by itself?

A. No, that's not possible.

Q. Why is that, sir?

A. Well, the Key Reader would have no idea which C within the menu trees to go to. There are many of them.

Q. How does the Key Reader know which C to go to?

A. It relies on the order in which the letters are presented to it and its understanding of the hierarchy of menus.

MR. REBACK: Your Honor, may—I understand the Court's objection. I would now like to ask a question for the purpose of a proffer. I don't mean to quarrel—

THE COURT: Go ahead.

Q. Is there any way to make a translator without embodying the hierarchy?

A. No. The hierarchy—the order involved in a computer language, its syntax, is an integral part along with the words or vocabulary that are used. That's just basic computer science. **[4-143]**

THE COURT: Well, you know, I don't need expert testimony to know that, do I?

MR. REBACK: No, sir. You said it before this case.

THE COURT: Well, apart from what I've said. I mean, it's a matter of common sense once you understand what we're talking about.

MR. GUTMAN: If the point is, your Honor, there somewhere has to be something in the program that helps it interpret, then absolutely. If the point is, as will be argued later, they had to do it this way, then it's clearly not. And this is a question and answer beyond the scope of the stipulation.

THE COURT: Well, I'll allow the question and answer to stand. I don't think it takes any expert testimony to tell me that's so, and so we don't need to waste time on it. Go ahead, next question.

MR. REBACK: Yes, sir.

Q. I believe Mr. Roshfeld earlier today demonstrated a macro slash WIR. Do you recall seeing that, sir?

A. Yes, sir, I do.

Q. Could you use the menu editing feature to demonstrate that macro, how the Key Reader would deal with that macro.

A. Sure. Basically what would happen is that as the macro executed, those keystrokes would be taken from the spreadsheet and presented to the Key Reader one by one. So first it gets a W of the WIR and it scans down this first column of information. And let me just—okay. Now it looks more like a first column. This is all format. It scans down the first column of information until it finds a W. In this case, the very first position we're in has the menu key W which I'm pointing to with my mouse cursor. **[4-144]**

Okay, fine. Let's go to the next level. If I press my plus key, I can then see the next level. I move on to the next level. The next keystroke presented to the Key Reader is then an I, so it scans down looking for an I. That happens to be the second menu key on the list. So we'll hit plus again and move into there.

Now it's got one letter left. It's looking for the R. The first one is a C, the second one is the R, and it knows it's reached

its appointed destination, no more characters. And we can see from this area of the screen called actions that we have identified a menu equivalent for WIR and that equivalent is row insert.

Q. So for the purposes of the record, where you're holding the cursor now is the menu equivalent, sir?

A. That's correct.

Q. Are those 1-2-3 commands, sir?

A. No, menu equivalents are the Quattro Pro native macro **[4-145]** language.

Q. For purposes of completing the demonstration here, cursor back up to the W portion.

A. Sure. Okay.

Q. Is there a menu equivalent for that?

A. There's not yet enough information available to the system for it to have found a menu equivalent. You can see only the general part of this is filled out. It's called basics. Since we're looking at W and that has to do with the worksheet, that's pretty basic to spreadsheets, but we don't know enough yet. We haven't gotten any more letters to know what—that is blank. There is no menu equivalent for this position.

Q. Okay. Now, sir, I'd like to take you out of the menu editor and bring you back to the display of the spreadsheet of Quattro Pro DOS Version 4.1.

A. Okay. I've just clicked on exit and we're now back in the spreadsheet displays.

Q. Incidentally, sir, to save us time here, is this menu editor—could you also have demo'd it for Quattro Pro 2.0?

A. Sure.

Q. 3.0?

A. Sure.

Q. Now, Mr. Roshfeld demonstrated in his demonstration that menus are sometimes displayed when the Key Reader is **[4-146]** running. I presume that's the case, is it not, he showed us?

A. Yes, that's correct.

Q. Okay. Could you bring up—I believe he used the macro RFC. I believe I saw that.

A. Yes.

Q. Actually, I guess the better one that he demonstrated for this purpose was WGF. No, excuse me. Excuse me.

MR. REBACK: This is a part that we're trying to accommodate your Honor and I'm a little ragged on this.

Q. RFC is the one that he used, but if you would edit that macro to leave it interactive, but taking away the C so that we could see that menu as it came up.

A. I don't think I have the diskette.

Q. I have the copy that was given to me by Mr. Gutman, and I'm now going to give it to you.

A. Okay. Let me insert this into the diskette drive. I'll click on file, retrieve, and I guess this must be on drive A. So it's called demo. I'll click on demo.

Okay. Here's the file, and I think the RFC macro you're talking about is this one that I've moved the cursor to.

Q. Yes.

A. Okay.

MR. REBACK: Frankly, your Honor, we don't **[4-147]** have the menu command hierarchy memorized here, so we're having a little trouble here.

Q. Can you get me back—display the menu that Mr. Roshfeld did, or would it be easier using the WGF macro? We know that will display it.

A. I can try with this macro. If you'd like me to edit it, I'll go ahead.

And I'm just going to backspace over this C, and then if I put—then put in the question mark at this point, I think that if we execute this macro, it will bring up the menu you're looking for.

So I'll go over to the tools menu and I'll select tools, I cursor down to macro, click. Down at the bottom we have the execute command. I click there. It's asking me which macro to execute. The cursor is already there, so I'll just click on enter, and I think this is what you're looking for here.

Q. Yes, sir, it is. Now, is that menu a Quattro Pro native menu?

A. Yes, it is.

Q. Can you get out of the macro and show me that same menu in the Quattro Pro native menu tree?

A. Sure. I'll just press control break to take the macro away. I'll move up to the style selection numeric format, and here you have the same menu.

[4-148] Q. Yes, sir. Thank you, sir.

Get us back up to the main menu bar, if you would, please.

A. Okay.

Q. I believe that Mr. Roshfeld also demonstrated the graph—the graph-type submenu. Can you pull that up in the native menu tree?

A. Sure. I just come over to graph, click on graph with my mouse, go to graph type, click again, and here we have that menu.

Q. Okay. Now, when Mr. Roshfeld showed it to us he showed us a pop-up box the same way we saw the other format menu.

MR. REBACK: And just for the record here, your Honor, it didn't have the pictures. It just had the words.

Q. Mr. Warfield, I take it since we're demonstrating this in the native mode—that's a native menu, is it not?

A. That's correct.

Q. Yet that menu—and that menu would appear when the Key Reader is running, would it not, sir?

A. That's correct.

Q. Are some of the commands in this submenu the same as the commands in Lotus 1-2-3?

A. Yes, they are.

[4-149] Q. Why is that?

A. From my perspective—

MR. GUTMAN: Judge—

THE COURT: Wait a minute. Sustained.

MR. REBACK: Give me just a moment, sir.

Q. Could I hear the question back just for the purpose.

THE COURT: Why is that? Why is that.

MR. REBACK: I'm sorry the one before that I lost track with the objection.

THE COURT: All right.

MR. REBACK: Give me just a moment, sir.

THE COURT: All right.

(Defense counsel confer.)

MR. REBACK: The objection to that question was sustained?

THE COURT: Yes.

Q. Will Quattro Pro DOS that we see on the screen here respond to a user simply by typing 1-2-3 keystrokes?

A. No.

Q. Earlier today I saw Mr. Roshfeld typing 1-2-3 keystrokes in the context of the macro. How can you reconcile your answer with what he showed the Court?

A. He typed some keystrokes—he seemed to have it pretty well memorized or something, because he was typing in empty boxes at one point and I don't really know what he **[4-150]** did, but he was running in a macro and those were macro keystrokes, and, you know, it wasn't what I'd call a 1-2-3 keystroke.

Q. If you could just take us back up to the main menu bar.

A. Sure.

Q. Without the Key Reader running, if we just tried to get our 1-2-3 keystrokes, would we be able to do that?

A. No.

Q. Can I write a 1-2-3 macro using the Key Reader, sir?

A. No, the Key Reader does not provide any assistance in writing macros at all. It's just like a phonograph. It just plays them back.

Q. Can I write 1-2-3 macros using Quattro Pro at all?

A. Well, you can sort of type in the characters here as we've done in some cases, but you wouldn't have any way to really tell what you were doing without any way to refer to what the menus were.

Q. In Mr. Roshfeld's demonstration I noticed that in Lotus 1-2-3 the clock is displayed on the screen. Do you remember seeing that, sir?

A. Yeah, I think so.

Q. Could you write a macro using the 1-2-3 commands to display the clock on the Quattro Pro screen?

A. I wouldn't really have any idea how to go about doing **[4-151]** that.

Q. Why is that?

A. I don't have the menus available to me so I can see which keystrokes are expected to do that in a 1-2-3 compatible way. I could certainly do it for our native menus.

Q. Suppose Mr. Roshfeld wrote such a macro and gave it to you and you wanted to edit some of the macro keystrokes high up in the macro. Would you be able to do that?

MR. GUTMAN: Object. Your Honor, now it's a hypothetical to an expert who's not supposed to be an expert.

THE COURT: How do you respond to that objection?

MR. REBACK: Well, I am trying to demonstrate the capabilities of the Key Reader. Since the Key Reader is for the most part invisible, I can only demonstrate those capabilities by asking the witness what it can do or can't do. I have no other way to demonstrate the capabilities. I'm not asking him as an expert.

THE COURT: Well, then I'm not getting the benefit of the demonstration. I'm getting the benefit of testimony, not a demonstration.

MR. REBACK: Well, your Honor, what the product can't do is an important part of the case, we would **[4-152]** respectfully argue. I can't demonstrate that except asking the witness to do it and having him tell me he can't.

MR. GUTMAN: The current question, your Honor, doesn't ask about what the product can do. It asks about what this witness can do and based on his personal—

THE COURT: I think there's a point there. I have seen a demonstration earlier of what the product can do dealing with the very point that you're now raising, it seems to me. If I go back to the preceding question, all one would need to do in order to use this program would be to have a Lotus 1-2-3 menu sitting there beside him as he worked with this program.

MR. REBACK: I think that's probably right.

THE COURT: Well, so you're not demonstrating something the product can't do. You're demonstrating, I suppose—one of the captions here under specific makes the point. If I'm remembering correctly, this was not something that was reported on the record but something I saw on the screen under specific, the caption hide, and what you're now demonstrating for me or seeking to demonstrate not through the demonstration but through questions is that the capabilities of this product are deliberately hidden.

MR. REBACK: Oh, they're not deliberately hidden, your Honor, not by any means.

THE COURT: What was the meaning of that **[4-153]** caption hide up there?

MR. REBACK: I don't even know what your Honor—

THE COURT: Does the witness know what I'm talking about?

BY MR. REBACK:

Q. Is there a menu command hide?

A. Could it be in the Windows product? There is a hide under the redraw suppression.

Q. It hides the cells. Mr. Warfield, it hides the cells, is that correct?

A. No, it merely suppresses unnecessary display during macro execution. I think Mr. Roshfeld mentioned that they have a feature like that in 1-2-3 as well.

MR. REBACK: I don't think that has any relevance to what we're talking about here. I mean, if your Honor is going to take the position or has in his mind that we have deliberately hidden certain capabilities—

THE COURT: I haven't taken any position. I'm trying to hear evidence and understand it.

MR. REBACK: Okay. We have found the word hide. This is in the Lotus 1-2-3 menu tree and I believe it is under the column command, which is under the worksheet command.

MR. GUTMAN: There is a command, your Honor, **[4-154]** that allows you to hide a column of data. I'm not sure at what point in the demonstration your Honor saw it. I'm not sure.

THE COURT: I'm talking about this demonstration, not the demonstration I saw this morning, and it occurred in this demonstration at a point where you were showing me that the specific meaning of the pluses, for example, and of the execution was not revealed anywhere on the screen.

MR. REBACK: It's not in the program anywhere.

THE COURT: Well, you know now that we've talked about it. Take it back. I won't require you to interrupt what you're doing, but we'll come back to it if you want.

MR. REBACK: Very well, sir.

BY MR. REBACK:

Q. Are you familiar with Quattro Pro for Windows, sir?

A. Yes, I am.

Q. Okay. Could you bring that up on the screen?

A. Sure. Let me exit Quattro Pro for DOS and I'll double click on Quattro Pro for Windows so the Windows program will then start up.

And so here we have it.

Q. What was your role in the development of Quattro Pro for Windows, sir?

[4-155] A. I was also responsible for the development of Quattro Pro for Windows.

Q. Okay. I'm going to ask you to demonstrate, sir, the features.

First, what version are we looking at here?

A. You're looking at the most recent version, Version 1.0.

Q. Is this the same screen display as Quattro Pro for DOS?

A. No. As you can see, there are a quite a number of differences here.

Q. I believe Mr. Roshfeld made mention or access to these tabs along the bottom of the screen. Would you move the cursor there. What are those things?

A. Sure. This is a new feature we've come up with that uses these notebook-like tabs to allow you to step through a number of spreadsheets that are all contained within the same notebook file.

Q. All right. Now, if you move the cursor to the line under the top line menu, sir, could you tell me what that is?

A. This line here we call the speed bar. It contains a number of iconic buttons that a user can click on with the mouse in order to initiate commands.

Q. For example, let's move over to the scissors, picture **[4-156]** of the scissors there. What does that do?

A. The scissors here, as you can see by the message at the bottom of the screen, is a command where if you click it on, it will cause whatever cells you've marked over here in the spreadsheet to be moved onto the clipboard where they can then later be pasted back into the same spreadsheet or other files.

Q. Does Quattro Pro for Windows use a two-line moving cursor, sir?

A. No, it uses pull-down cascading menus with dialogue boses.

Q. I've heard that term mentioned. Could you show me a dialogue box, sir.

A. Sure. If we go over to the file menu, click on file, move down to retrieve, click on retrieve, now we're seeing what's called the dialogue box here.

MR. REBACK: Give me a moment, sir.

(Defense counsel confer.)

Q. Does Quattro Pro for Windows display the 1-2-3 menus?

A. No, it does not.

Q. Does it contain the 1-2-3 menus?

A. No, it does not.

MR. GUTMAN: Objection. It's calling for a legal conclusion.

THE COURT: Well, I sustain the objection to **[4-157]** that question.

Q. Can Quattro Pro for Windows run 1-2-3 macros? I mean, we saw it today, so I presume it can run some of them at least.

A. Some, yes.

Q. Also has a Key Reader feature?
A. That's correct.
Q. And could you take us out of the dialogue box—
A. Sure.
Q. —and run through the steps to demonstrate to the Court how you turn on the Key Reader.
A. Sure. To turn on the Key Reader, I just take the mouse up to the top where it says Quattro Pro for Windows, click the right mouse button. I am then presented with a dialogue box again. If I move over here to where it says macro and I click there, we see here a section entitled macro options. The Key Reader you can tell by this check mark is actually already on, so at this point we can just click okay.

MR. REBACK: Just a second, sir.

(Defense counsel confer.)

Q. Is the default setting for the Key Reader on or off?
A. I think by default the Key Reader is off.
Q. Now, trying to shortcut some of this, sir, in Mr. Roshfeld's demonstration using interactive macros with **[4-158]** Quattro Pro Windows, he caused some blank boxes to appear on the screen. I believe he used the graph-type menu to do that.

Do you think using his macros you could just present the same screens that he did?
A. Okay. Well, let's try bringing up his file and see how we do.
Q. You can put in another macro?
A. If we want to just bring that up, slash GT.
Q. Will that do that?
A. I think so.
Q. Okay. Well, give it a try.
A. Okay. So this is for graph type. We'll put a question mark. As I recall, they had earlier turned on the macro recorder. Not the macro recorder. Debugger. Select debugger.

We can see by this lower indicator in the lower right that the decoder is turned on and execute this macro by clicking

tools, macro, and then execute. We're already in the cell containing the macro, so I'll just click okay.

We then press the space bar to move through, and are these the boxes you're looking for?

Q. Yes. Do you have any idea—well, strike that.

Was anything supposed to be in these boxes?

A. Originally we had planned to have 1-2-3 compatible **[4-159]** menus in these boxes.
Q. Why aren't there any menus in those boxes now?
A. Well, we decided to take the menus out in the summer of '92.
Q. Why do the boxes still appear?
A. It's a bug in the program.
Q. Well, if you could remove the command, why couldn't you just remove the boxes?
A. Well, actually it was pretty easy to delete all those words out of the file, but the code itself is quite complex. Making sure at each step that just the exact right thing was displaying under all possible conditions was more difficult. In fact, we didn't actually discover this bug until just shortly before we were ready to ship the product.
Q. Do you have any plans to fix this bug?
A. Yeah, we intend to fix it with the next release of the product.
Q. Is there any product documentation that tells you what if anything you can do with these boxes?
A. No.
Q. Is there any aspect of the on-line help feature that tells you what you can do with these boxes?
A. No, no. This is a bug. We don't document bugs.
Q. Could you cursor down in this box for me.
A. Let's try it. I think I'm now at a point where— **[4-160]** yes, I can cursor inside the box.
Q. Stop for a second. Could you tell me what command would have gone there?
A. Something to do with graph types. I can't be sure exactly where I am here.

Q. Okay. Could you press F1 for help and see if we can get some help as to where we are.

A. Sure. It looks like there's no help.

Q. So the screen says "Help not available"?

A. Yes.

Q. All right. Now, I take it that the user as a general matter can find information on the Key Reader through what, the help feature of the program?

A. All of the documentation of the program is a place where a user could go to find information on the Key Reader.

Q. Take me back to Quattro Pro 2.0, please.

A. 2.0. Let me exit here.

Okay. Quattro 2.0 is listed right here, so I'll double click, and we're in Quattro Pro for DOS Version 2.0.

Q. Now, you've mentioned various aspects of documentation for a program. What are the aspects of documentation for this program?

A. Well, you've got the printed materials that come with the program, you've got the on-line help file that comes with the program, and in addition there's a file with the **[4-161]** program called the read me file.

Q. What is a read me file, sir?

A. A read me file is just a file that's used by a lot of different software that contains addenda of various kinds, things that an end user wouldn't find elsewhere in the documentation that are late-breaking news that they need to be sure to check on.

Q. Could you demonstrate the read me file in this program, sir.

A. Sure. Probably the easiest way to do is that is, I'll select file, utilities, DOS shell, and we can see here by the prompt on the screen—I can't point to it—somewhere C hyphen backslash Quattro Pro 26. We're in the directory. And now if I just type read me, it brings up the read me file for Quattro Pro 2.

Q. Was this version of Quattro Pro the first version to have the Key Reader?

A. Yes, it was.

Q. Was the Key Reader mentioned in the documentation?

A. Yes, it was.

Q. Where is it, sir?

A. Actually it's mentioned in this read me file.

Q. Show me where. Show the Court where.

A. A couple of different places. If we cursor down, which I'm doing, you can see in the contents item number 21 **[4-162]** is macro Key Reader.

I can then use my page down key. I'll page down within this file to the point where we have item number 21, and I'll just bring that to the top of the screen, and you can see here this is the section on the macro Key Reader.

Q. Yes, sir.

MR. REBACK: Now, your Honor, we have previously marked this in hard copy, so I won't keep it on the screen unless your Honor is interested in seeing that.

THE COURT: All right.

Q. Is the macro Key Reader documented anyplace else in this program, like in the hard copy, for example?

A. No, it's not.

Q. Why is that?

MR. GUTMAN: Objection, your Honor.

THE COURT: Sustained.

MR. REBACK: Okay. I'm sorry.

Q. How would the user know to look at the read me file?

MR. GUTMAN: Object.

THE COURT: I'll permit that question.

A. Actually there's two different ways that users would know. Most experienced users expect to find read me files with software because most companies make a practice of providing such files. In addition, Borland provides a notice which I have here (indicating), it says "Read Me **[4-163]** First", that appears in all of our products directing the user to go to the read me file when they want to find something.

MR. GUTMAN: Objection, your Honor. Move to strike the answer. This is—first of all, in terms of what most users look

for, that's expert opinion. It's not within the scope of the stipulation. And again, this business of what's in the box or wasn't in the box is beyond the scope of the demonstration.

MR. REBACK: Well, I think.

THE COURT: Overruled.

MR. REBACK: Your Honor, if you just give me one more minute to pack things up.

(Defense counsel confer.)

BY MR. REBACK:

Q. Okay. Can you get me back into the menu editor, Mr. Warfield, so that we can find this hide command—it's in Quattro Pro DOS 4.0?

A. Let me try to go back exactly where it was. So it started up, 4.01. I believe we went to options, startup, edit menus and the menu editor, then read in the information.

Q. Okay. Are any menus being hidden here, sir, or concealed from view?

A. Well, we haven't expanded them fully, but you can get **[4-164]** them with plus and minus.

Q. Get me down into the Key Reader selection again down below the line, extend them out.

A. I'll hit plus and then cursor.

Q. Okay. Now, are any menus being hidden from view?

A. There's no menus, really, in the Key Reader.

Q. So the answer to my question is no?

A. No.

Q. All right.

MR. REBACK: Now, your Honor, in light of your Honor's prior rulings, maybe this is a proffer or summary exhibit. Let me just explain what I'm trying to do and your Honor can rule.

THE COURT: Go ahead.

MR. REBACK: I could have Mr. Warfield—it would take most of the afternoon—read into the record the order in which these pickletters appear or your Honor could just see them in chambers. We produced to the other side about three months

ago Defendant's Exhibit 520, which is the contents of the file, and I would like to simply have the witness authenticate that and say that the pickletters are in the same order as you see them on the screen.

THE COURT: Do you have any objection to that?

MR. GUTMAN: I'd like to take a look at what form. There are various things we were shown. All of them **[4-165]** were redacted.

MR. REBACK: No, this is not redacted. This is as shipped and this is exactly what we gave you. Let me place Exhibit 520 before the witness.

I should note for the record it is stamped "Confidential," your Honor, "Attorneys Eyes Only." May I now just ask the witness those two qualifying questions, your Honor?

THE COURT: Do you have any objection?

MR. GUTMAN: I'm sorry. Was the question whether I objected to this document?

THE COURT: No. State again what you wanted to—

MR. REBACK: I want to ask the witness if he will identify this as the human readable version of the Key Reader portion of the QUATTRO.MU file.

THE COURT: If you want him to do it through the demonstration and you can check it, that's fine.

MR. GUTMAN: I certainly don't want to waste the court time having him check it through the demonstration. I don't believe that one would actually see what's on this document if you were to do what Mr. Reback has threatened to have him do. And having him authenticate the document is I believe beyond the scope of the stipulation, but if he wants to do it, I will not object to **[4-166]** him doing it.

THE COURT: Now, I understand the purpose of what you're proposing is to do in a shorthand version of hard copy what would be shown by the demonstration. Mr. Gutman is telling me he has doubts that that is what the shorthand hard copy version would show if we went through the demonstration. I suppose if we can't get that resolved, we better just go through the demonstration.

MR. GUTMAN: If—if I might have a moment to consult with Mr. Reback, your Honor, we may actually be able to agree on something.

THE COURT: You may.

(Counsel confer.)

MR. REBACK: Mr. Gutman wants to cross the witness on this exhibit.

THE COURT: All right.

MR. REBACK: Let me just say—

MR. GUTMAN: I'll say that, your Honor. I don't have to do it.

MR. REBACK: Let me just say that for the purposes of the record I'm not going to take time to march down through there because the computer program is in evidence and any ordinary observer could cause these things to appear. So I think for the purposes of my argument I'm fine and so I need not belabor the Court on the point.

[4-167] (Defense counsel confer.)

BY MR. REBACK:

Q. Could we get—do you have any idea where the word hide or hidden—

THE COURT: I'll tell you where it appears. It appears where specific now is and there's a blank. Hide shows up in there at certain points. What is it that that means? Whether it's a menu, command or legend or explanation or whatever it is, what is it?

MR. REBACK: I think it's a menu equivalent.

Q. Can you get it up on the screen, Mr. Warfield?

THE COURT: I think it came up when you were demonstrating some of the macros.

Q. It's under column, Mr. Warfield. Can you find it?

A. Well, which one is column under? Do the Lotus guys know?

Q. I think it's off the worksheet command, the fourth one down, I think.

A. C?

Q. Expand it out, could you? Down one.

A. Hide.

Q. There we go.

THE COURT: Now, what is the function?

Q. What is that hide, sir?

A. It's just what Mr. Gutman was talking about, the same **[4-168]** command where you have the ability to take a column—these are general actions that take place on columns. That's why it says general up there. And you can hide it so you can make the column disappear from the screen.

Q. Is that a menu equivalent?

A. Yes, column hide—

THE COURT: Let me understand. When you say menu equivalent—

MR. REBACK: Yes, sir.

THE COURT: —you have been making a careful distinction throughout this demonstration between the menu line and—what did you refer to it as, the next line that had buttons?

MR. REBACK: Speed bar.

THE COURT: Is the speed bar also a menu equivalent?

MR. REBACK: No. As Mr. Warfield testified, the speed bar is not part of the menus in any respect.

THE COURT: Is it a menu equivalent? Hide is not part of the menu in any respect either, is it?

MR. REBACK: No, sir.

THE COURT: Why do you call it a menu equivalent?

MR. REBACK: All right.

THE COURT: I'm trying to understand your **[4-169]** terminology. You see, you're speaking to me with words that you're using in a specialized meaning.

MR. REBACK: Yes, sir.

THE COURT: And I don't know what that specialized meaning is and I'm trying to understand it.

MR. REBACK: I understand.

THE COURT: Why wouldn't I, thinking in a broader sense in the way we ordinarily talk about menus, think of the speed bar as a part of the menu or a menu equivalent even though it's not a part of the menu as you're using the term in a narrower sense than we ordinarily in talking about software programs use the term menu?

MR. REBACK: Okay. I should not have demonstrated the speed bar, because it's irrelevant to anything we're talking about here, but I do want to talk about the difference between menus and menu equivalents.

Menus are what you see on the screen, so in this product you can only see the Quattro Pro native mode. However, those menu commands are not processed in any respect by the software. They are translated into Borland's unique menu equivalents and then they are processed.

THE COURT: And that's processing in any respect? That's translating and then—

MR. REBACK: It is translating, sir, yes, sir.

THE COURT: All right. Now, let me see if **[4-170]** I've understood you. Where first a translation occurs, then you may speak of that, the thing that was translated, as a menu equivalent.

MR. REBACK: What it was translated to is a menu equivalent. What it was translated from was a keystroke, for example, or a menu command.

THE COURT: All right. I now understand the sense in which you're using it.

MR. REBACK: Yes, sir.

THE COURT: That doesn't mean that that's the only sense in which the word can be used or that it's the only sense that's relevant to copyright law. You understand, I'm simply trying to understand the evidence before me and I think I now understand the way you're using the term menu and menu equivalent, and you're using both of them as excluding other things that show up on the screen somewhere that help you know what to do to get the machine to function. And so in that respect, then, you use the term speed bar as not a part of the

menu in any sense, but it's also correct that it appears up there and one can click the mouse on it and make the thing function just as one can click the mouse on something in the menu line and make the machine function.

MR. REBACK: Yes, sir.

THE COURT: All right. I understand.

[4-171] MR. REBACK: Just one moment.

(Defense counsel confer.)

MR. REBACK: I've completed my examination, your Honor.

THE COURT: All right. Did you want to cross-examine?

MR. GUTMAN: I do, your Honor. I was under the impression that your Honor had another matter on, and perhaps if I could inquire as to our scheduling constraints, if any.

THE COURT: None. I don't think I have another matter on, as far as I know. Do I?

MR. REBACK: No, sir.

MR. GUTMAN: Your Honor, if I might have one moment.

(Plaintiff's counsel confer.)

CROSS-EXAMINATION

BY MR. GUTMAN:

Q. Mr. Warfield, when you were showing this part of Quattro Pro 4.01 to the Court, I believe you testified that this Key Reader file contains the embodiment of the hierarchy that's needed. The hierarchy, the menu hierarchy you're referring to in terms of the Key Reader is the 1-2-3 menu hierarchy, isn't that right?

MR. REBACK: Excuse me. Wasn't the question **[4-172]** objected to? Did I get that one in?

THE COURT: Read me back the question.

(Question read back by the court reporter.)

MR. GUTMAN: I don't believe it's the same question that was stricken, but I can make it clearer.

BY MR. GUTMAN:

Q. You referred to a hierarchy, Mr. Warfield, in your testimony concerning the Key Reader and the outline form there. The hierarchy you're referring to is the hierarchy of Lotus 1-2-3 menus, isn't that right?

A. That's correct.

Q. Now, can you get out of this and escape back to the main screen, please.

A. Sure. (Pause.) Okay.

Q. Now, when you were trying to show the Court before how to get from the WYSIWYG mode, which this is, to character mode, you went to the options menu.

THE COURT: Now, let me take note here. You have referred to this as character mode throughout your demonstration this morning and now. It was referred to by Mr. Reback as text mode, but I understand you both to be talking about the same thing.

MR. GUTMAN: I understand it—

THE COURT: If I have misunderstood, I wanted that clarified.

[4-173] MR. GUTMAN: No, I appreciate that.

MR. REBACK: No, there's no distinction, but it is currently in graphics mode.

MR. GUTMAN: Which is WYSIWYG. Graphics and WYSIWYG are synonyms for these purposes as are character and text.

Q. Now, Mr. Warfield, in order to get into character or text mode out of WYSIWYG, you went to the options menu and then you found various commands and subcommands under there, isn't that right?

A. That's correct.

Q. There's an easier way to do it, isn't there, sir?

A. Not that I'm aware of.

Q. What happens if you push the character button, the one next to—

A. You mean click on it with the mouse?

Q. Yes.

A. Go to character mode.

Q. That's what I thought.

Now, in terms—

THE COURT: Well, let me see now. Do you now agree that was an easier way to do it?

A. Considering I didn't think of it, it wasn't easier for me.

(Laughter.)

[4-174] Q. Let's leave ease out of it. You'll agree with me, sir, that that is an available alternative to going to the options menu, right?

A. Yes, sir.

Q. Does the same thing by clicking on the speed bar, right?

A. Yes.

Q. Now, you saw Mr. Roshfeld's demonstration this morning. You have had his disk to work with or study or do whatever you wanted during the lunch break, which was extended.

Any of the macros that he showed this morning, would any of them have worked any differently if he'd done his demonstration in WYSIWYG mode rather than character mode?

A. I don't think so, but I haven't studied them that extensively.

Q. Now, the buttons on the speed bar you referred to don't appear, or do they, in character mode?

A. Yes, these are the buttons over here in character mode.

Q. And there's a—okay. So they're both in character and WYSIWYG mode, right?

A. Correct.

Q. Now, am I right, sir, that the format—

[4-175] THE COURT: Excuse me for a moment. Is the one difference that the button that gets you quickly to character mode is not over here on the right? Those are already there, is that right?

THE WITNESS: I think it's available if I click on the bar here. There simply is not the same amount of room. So if I click, you can get to it, but you're—

THE COURT: You're already there.

THE WITNESS: You're quite correct, your Honor.

THE COURT: Otherwise you're saying that what's in the column on the right is the same thing that was on the speed bar.

THE WITNESS: That's correct. It's just truncated so we present it in two pages, if you will.

THE COURT: All right.

Q. And you saw that character and WYSIWYG were available when Mr. Roshfeld was demonstrating 3.0 this morning, right?

A. I don't have a specific recollection of that, but yes.

Q. Now, I believe you testified that insert and row, which were menu equivalent commands, weren't commands in Lotus 1-2-3. There are such commands in 1-2-3, isn't that right, Mr. Warfield?

MR. REBACK: Excuse me. May I hear the question again?

[4-176] THE COURT: Yes. It may be read back.

(Question read back by the court reporter.)

A. My recollection was that my testimony was that the menu equivalent insert row does not appear in 1-2-3.

Q. But there's a command under worksheet, insert row, isn't that right?

A. Yes, couple of menu picks there, definitely.

Q. Now, if I heard right, when Mr. Reback asked you whether you could write 1-2-3 macros using the Key Reader, you said no. You could write 1-2-3 macros using these products, though, can't you?

A. If you knew what to do, I'm sure you probably could. You can edit text in all of them.

Q. You can create using the same sequence of characters in Quattro Pro 4.01 and in Quattro Pro for Windows a 1-2-3 macro, isn't that right?

A. If you give me a set of characters such as slash WIR and type them in, I can type it.

Q. In fact, without being prompted you did it, didn't you, in Quattro Pro for Windows when you were trying to recreate a macro, and you created a 1-2-3 macro that began with the apostrophe and then had the slash and then G and T and a question mark within curly braces? While we were all watching, you created that in Quattro Pro for Windows, is that right?

[4-177] A. Actually, in a sense I was prompted. I'd been looking at slash WIR all morning in Mr. Roshfeld's field, but yes, I typed that.

Q. And that was a 1-2-3 macro, right?

A. Yes.

Q. Now, I got the impression listening to your testimony about how Mr. Roshfeld used these keystrokes in order to get commands to work in Quattro Pro that you are not terribly familiar with the 1-2-3 menu commands, is that right?

A. Well, I'm not sure just how you quantify terribly familiar, but I can tell you I don't have the whole command hierarchy memorized.

Q. But you have written programs based on it in your career, have you not?

A. Yes, I have.

Q. You wrote a program called Farsight, isn't that right?

MR. REBACK: Objection, your Honor. Beyond the scope of the direct.

THE COURT: Well, I don't think so. The objection is overruled.

Q. You wrote Farsight, correct?

A. Yes, I did.

Q. And Farsight was a clone of Lotus 1-2-3 Release 1A, isn't that right?

MR. REBACK: Objection. Irrelevant, beyond **[4-178]** the scope of redirect, assumes facts not in evidence.

THE COURT: You know, testimony that bears on evaluation of any of the answers given on direct is appropriate cross-examination subject only to 403 problems and that kind of thing. If seems to me I should allow this question if not for

other reasons, as demonstrated a few moments ago with respect to different ways of getting to character or text mode quickly and easily. I think I should allow this question.

MR. REBACK: For the record, Farsight is not even a Borland product.

MR. GUTMAN: We'll stipulate to that, but this man wrote it.

THE COURT: The witness has testified in a number of ways during direct to a lack of familiarity with the 1-2-3 menu structure. This is testing the credibility of those answers, as I understand. That seems to me to be perfectly appropriate to cross-examination even given the very limited scope of the direct under the stipulation here.

BY MR. GUTMAN:

Q. Mr. Warfield, isn't it a fact that Farsight was a clone, or a workalike, or menu, macro and keystroke compatible, pick your term, with Lotus 1-2-3 Release 1A?

A. Farsight was macro compatible with Lotus 1-2-3 Version 1A.

[4-179] Q. And the menus were basically the same as 1-2-3 1A, right?

A. From a macro perspective, that's true, yes.

Q. From a user's perspective that was true too, wasn't it?

A. It wasn't true to me. I really couldn't say about any other users.

Q. What did you see on the screen when you ran Farsight?

A. I saw pull-down cascading menus and I saw dialogue boxes.

Q. What words did you see, sir, in what order? Was the first menu: worksheet, range, copy, et cetera?

A. Actually, as I recall, the first words were WKST. There were a number of other words scattered throughout the menus. There were a number of words that didn't ever appear in 1-2-3. There was a lot of different menus in the product.

Q. When did you write that program?

A. We started on that program back in 1983.

Q. And do you remember what the menus were, what the commands were on that menu?

A. No, but I remember in general how they were constructed.

Q. You remember the first? What was the second item?

A. I don't recall.

[4-180] MR. REBACK: Your Honor—

THE COURT: On 403 grounds I sustain the objection.

Q. Now, didn't you also create Surpass?

A. That's correct.

Q. And wasn't Surpass a workalike to Lotus 1-2-3 Release 2.0?

A. I don't know what you mean by the term "workalike."

Q. Was it macro compatible?

A. I'll say what I said before. Yes, the product was macro compatible.

Q. And didn't you write—didn't you write the long prompts for Quattro Pro without looking at Lotus 1-2-3? Isn't that what you've testified in this case, sir, based on your knowledge of our product?

MR. REBACK: Objection.

THE COURT: Now, wait a minute. Mr. Gutman, let me remind you of the limited purpose for which I'm allowing these questions, so I am not going to permit you to argue to me that I should consider these questions and answers on any other issue than the credibility of the answers of this witness on the direct, not for any other purpose in this case.

MR. GUTMAN: That is the only purpose, your Honor.

[4-181] THE COURT: Well, we're rapidly approaching the point where I think for that purpose it's cumulative, taking more time than it's worth and so on 403 grounds, the value of its probative weight for the legitimate purpose of impeaching the credibility of the answers given on direct is diminishing rapidly and is very likely to be overcome so that I should sustain the objection on 403 grounds.

MR. GUTMAN: I just have one more question on this line, your Honor.

Q. Mr. Warfield, isn't it the case, sir, that you wrote the long prompts of Lotus 1-2-3—I'm sorry—of Quattro Pro for the 1-2-3 compatible menus, and by your own testimony you did it without reference to Lotus 1-2-3?

MR. REBACK: Your Honor, I really am going to object to that. Let me explain why.

THE COURT: I sustain the objection.

MR. REBACK: Thank you, sir.

Q. Can we get back to Quattro Pro 2.0?

A. You'd like me to exit and go to Quattro Pro 2.0.

Q. Yes.

A. Okay. (Pause.) Here we are in 2.0.

Q. Now, what command—is there any command on this menu that would show you the read me file without going to DOS?

A. No, sir, there is not.

Q. Now, you showed that there was a reference to the **[4-182]** macro Key Reader in the read me file. Is there any reference to it in the printed material, sir?

A. No, there is not.

Q. Nothing in the manual?

A. Not that I'm aware of.

Q. Nothing on the cover of the box that the program comes in where features are listed?

A. I don't think so.

Q. Are you familiar with the panel?

A. Very much so, yes.

Q. On the back of the box where it lists new features, nothing on there about Key Reader, right?

A. I have no recollection of anything.

Q. How about in help? Is there any reference in the help?

A. I don't think so.

Q. Why don't you activate the Key Reader command and let's see—let me back up.

This program has context sensitive help, doesn't it?

A. By that you mean what?

Q. By that I mean that when you ask for help, the response is sensitive to where you are in the program at the time you ask.

A. Yes, that's correct.

[4-183] Q. Fine. Now, let's go to the macro Key Reader command and see what help says when you ask for it.

A. Okay. So I'll click on tools, I'll click on macro, I'll come down here to Key Reader. Do you want me to click on that, I presume, and then press Fl for help?

Q. Yes, sir.

A. Okay. Here we are.

Q. Any reference to the Key Reader in there?

A. Let me read it. Can't see any reference to the Key Reader.

Q. Okay. Now, the Key Reader wasn't a bug, was it?

A. No.

Q. You expressed some uncertainty as to whether there was any mention of Key Reader on the box.

A. Yes.

Q. Would you like to see the box? Let me refresh your recollection.

A. Okay.

Q. Can you point out to me any reference to the Key Reader in the materials on the box?

A. Let me just read through.

I can find no explicit reference to Key Reader on this box.

Q. Was there any reference to it on the box for the 3.0 release of Quattro Pro?

[4-184] A. I don't recall.

Q. Okay. Do you have Exhibit 520 in front of you, sir?

A. Yes, I do.

Q. Could you explain to me, please, what the parentheses mean on that exhibit?

A. The parentheses on this exhibit are simply the—

MR. REBACK: Excuse me, your Honor. This is marked "Confidential - Attorneys Eyes Only." Could we—the way we manipulate menus is a confidential aspect of the company's business. Could we take this at side bar or some other way if it's necessary?

THE COURT: Well, how do you propose we handle it?

MR. REBACK: I'm willing to let the witness testify about it. I just don't—

THE COURT: There's not a problem with anything that's on the screen?

MR. REBACK: No, sir.

THE COURT: All right. I'll see you at the side bar over here.

[4-185] AT SIDE BAR

THE COURT: It probably would be better if you stood facing from that direction, whoever is going to speak, so that you're speaking toward the court reporter.

MR. GUTMAN: Okay.

BY MR. GUTMAN:

Q. What do the parentheses mean?

A. They're just a formatting command, just as—you see this is an indented outline type of format. We found it easier rather than trying to count how many spaces there were from a formatting standpoint just look for a parenthesis to indicate the next level down.

Q. So the parentheses indicate different hierarchical levels?

A. Yeah. I mean, it just tells it to go to the next level.

Q. Well, now, a parenthesis facing to the right indicates what?

A. Open parenthesis means go one level deeper in the outline.

Q. And close parenthesis means what?

A. Come back out to the level you were last at.

Q. So then am I right in understanding that the letters NCRAMBI would refer to a level of the menu hierarchy?

A. That's correct.

[4-186] Q. So that would be a screen submenu, for example?

A. If you view it as a menu, yes.

Q. And are there commands from the 123.MU file that are not represented on this collection of sheets?

MR. REBACK: I'm sorry. Is that the question you intended to ask?

MR. GUTMAN: Yes.

MR. REBACK: Can I hear that back?

THE COURT: It may be read back.

(Question read back by the court reporter.)

MR. REBACK: This was represented as the Key Reader portion of the QUATTRO.MU file. That's all we've tendered it as, so I don't really understand the question.

MR. GUTMAN: But the witness has already testified at deposition and indeed we already have in the record that the Key Reader file was created by editing the 123.MU file.

MR. REBACK: That's not in the record yet. If you want to move it in tomorrow, perhaps—

THE COURT: Well, are you suggesting we should bring this witness back for—

MR. REBACK: Of course not. I'm sorry.

BY MR. GUTMAN:

Q. So what I want to know is whether this is complete or whether there are commands reflected in the 123.MU file that **[4-187]** are not reflected in these sheets.

MR. REBACK: The problem I'm having here is that the 123.MU file includes all the native menus. It's a huge file. So that's my confusion. You already know that.

THE COURT: Well, I don't know the answer to the question, so I'd like to have the answer to the question unless you can persuade me it's not appropriate for some reason.

Wait a minute. Did you wish to be heard further?

MR. REBACK: No, sir.

THE COURT: All right. You may answer the question.

THE WITNESS: Maybe I better just hear it back again.

(Question read back by the court reporter.)

A. Okay. There's a lot of information in 123.MU that's not in the Key Reader. What do you mean by commands?

Q. Okay. Let's back up. The pickletter W refers to the worksheet command in 1-2-3?

A. In 1-2-3 it does, you're correct.

Q. And the pickletter G, which is according to the parentheses indented one level in the menu hierarchy from worksheet would refer to global, would it not, which appears as the first command under worksheet in the 1-2-3 menu hierarchy?

[4-188] A. I see. So you're going to refer to any position within this hierarchy as a command.

Q. Please answer the question, Mr. Warfield. The question is: Doesn't G refer to global?

A. Yes.

Q. It represents global. And global is the first command under worksheet in the 1-2-3 menu hierarchy, isn't that right?

A. I think we have a terminology problem. It is difficult, the choice there. Is it a command? No, it doesn't do anything.

Q. It is the first word under worksheet?

A. Yes, absolutely.

Q. Now, what about format? Isn't that—isn't the parentheses and the F, pickletter F under the pickletter G, indicative of format, which is the first word you would find under global, which is in turn under worksheet in the 1-2-3 menu command hierarchy?

A. Yes.

Q. Now, where would one find, if anywhere, in this file letters that would correspond to the submenu that appears under format in the 1-2-3 menu hierarchy?

A. Okay. That's not part of the Key Reader portion. That's in the menu equivalent portion of the file.

Q. And why is that?

[4-189] THE COURT: And that means it's not in Exhibit 520?

MR. REBACK: That's correct, your Honor.

THE COURT: Well, all right. I understand the point.

MR. GUTMAN: That is—our whole point, your Honor, is that this is incomplete.

MR. REBACK: You would not let me ask the question why, that very same question before.

THE COURT: Well, I may let you ask it now. Go ahead.

MR. REBACK: May I ask a couple of other questions to set a predicate for it?

MR. GUTMAN: He didn't answer if I asked why. If his own—

MR. REBACK: Is this document complete?

THE WITNESS: Absolutely. This is the entire contents of the Key Reader portion of the file.

THE COURT: Well, now, you see, it's complete for what it is, but it does not—you have deliberately by definition excluded from it some very interesting and important and significant things. So I understand the answer it's complete, but when you start arguing to me that it's complete for the purposes of what I need to think about in order to make my rulings in this case, the answer is it **[4-190]** is not complete from that point of view.

MR. REBACK: Your Honor, that's the reason I wanted to ask him about this in direct.

THE COURT: Well, I'm giving you the privilege of asking right now if you want to.

MR. REBACK: Okay. Where are the format subcommands that the Key Reader would refer to?

THE WITNESS: They're within the menu equivalents.

MR. REBACK: Where are the menu equivalents?

THE WITNESS: They're in the menu equivalent portion of the MU file.

MR. REBACK: Are these format subcommands Lotus commands or are they Quattro Pro—

MR. GUTMAN: Objection.

THE COURT: Well, that's all right. I will let it come in. It doesn't answer all the questions that are relevant to my decision making, but I understand the answers that are being given.

MR. GUTMAN: As long as the point is clear that this is not the whole thing.

MR. REBACK: When the format command is invoked, will it call in the format subcommands from other portions of the file?

THE COURT: Wait a minute. I didn't hear the **[4-191]** first part of the question. When the format command is what?

MR. REBACK: Invoked. When invoked, will it call in the format subcommands from other portions of the file?

THE WITNESS: Sure, because at that point you've translated that F into a menu equivalent. This is the syntax for it right here that would appear in a macro defaults format. It then goes into the menu equivalent portion of the program and activates a default format command.

BY MR. GUTMAN:

Q. So format is now a command?

A. Default format is, yes.

Q. Where is the submenu for item L, which is the next item in order?

A. Same story. It's in the menu equivalent portion of the file under the defaults alignment command.

Q. Okay. Now, on page 572561, am I correct in understanding pickletter R there is range?

A. Yeah, I guess so.

MR. REBACK: Don't guess.

A. I couldn't tell.

Q. We can get the menu tree chart or you can look at the notation block.

[4-192] A. That's right.

Q. Block is what Quattro calls range in its native menus, isn't that right?

A. Sure.

Q. They're synonyms, right?

A. Sure.

Q. Okay. Now, where is the range format submenu?

A. It's in the block format menu equivalent section of the file.

Q. Now, is that submenu the same in 1-2-3 and in Quattro Pro?

A. I think so, yes.

Q. Now, is that the same as global format or is range format different?

A. The menu or the command?

Q. The submenu.

A. The submenu has the same choices on it.

Q. Okay. Now, where is the submenu for label under range? That's what the L refers to, isn't it?

A. Yeah, right below here. It's right here in the block align menu equivalent portion of the file.

Q. So it's not in the these pages, it's in the menu equivalent portion?

A. That's correct.

Q. Okay. Where would we find file erase?

[4-193] A. Same place. It's in the menu equivalent portion of the file.

Q. Is there any indication on here?

A. Yeah, there's a menu equivalent printed right to the right of—

Q. That's block erase. How about file erase?

A. Kind of hard to tell without the context. Do you know which page that's on?

Q. Well, file, I believe, is on 572564, pickletter F.

A. Okay. You want file erase, is that correct?

Q. Yes.

THE COURT: Over on the next page. It's letter E.

A. Yeah, file erase on 6-5.
Q. Now, where is the submenu?
A. It's in the file erase portion of the menu equivalents.
Q. And file S, where is submenu?
A. Same place, file—
Q. Back on the preceding page, pickletter F for file, pickletter R for retrieve—
A. Okay.
Q. —where are cancel and replace?
A. They're in the menu equivalent portion of file retrieve.

[4-194] MR. GUTMAN: Okay. I'm done with that.

(Plaintiff's counsel confer.)

Q. Now, is the 123.MU file in the same order as this document?

MR. REBACK: I'm sorry?

Q. Are the contents of the 1-2-3.MU file in the same order as this document? The organization and outline of commands, is it the same as in this document?

MR. REBACK: Well, I object on the grounds that that's not relevant to anything. That document is in evidence already. If you just refer to that and compare the two documents—
THE COURT: You want me to do that instead of having somebody who's done it answer it for me?
MR. REBACK: I don't know that he's done it.
THE COURT: Well—
MR. REBACK: They moved it into evidence, your Honor. It's in evidence.
MR. GUTMAN: We didn't move it into evidence.
THE COURT: I'll overrule the objection.

A. So what's the question?

THE COURT: You may answer the question. The question may be read back or you may restate it. It may be read back.

(Question read back by the court reporter.)

[4-195] A. So you're asking whether the outline of the commands in 123.MU is the same as the outline of commands found here?

Q. That's what I would like to know. We've already established that W means worksheet, G means global, F means format. L means label, et cetera, et cetera.
A. Right.
Q. What I want to know is whether they are listed here in this document, Exhibit 520, in the same order as they would be found in the 123.MU file.

MR. REBACK: Can we just bring the file over here? It's in evidence.

THE COURT: If he knows the answer, he may answer.

A. Yeah, as far as I know, they're listed in the same order.

MR. GUTMAN: Okay. I have nothing further on these documents, your Honor. If I may retreat to my desk, I may be done.

THE COURT: Unless you want to ask something that needs to be answered here at—

MR. REBACK: I need to make inquiry of the Court. This is the testimony I had tried to put on before, but I am willing to go into detail as to exactly how execution works so that your Honor doesn't come away feeling that we've taken something out of this file or concealed it.

[4-196] Just let me make a proffer first.
I proffer to show through the testimony—
THE COURT: Well, let me ask you how long is it going to take you to do it without going through the proffer first? How long will it take you?
MR. REBACK: I don't know, four or five questions maybe.
THE COURT: All right. Go ahead.

REDIRECT EXAMINATION

BY MR. REBACK:

Q. Does the format command in the document in front of you—I'm sorry, Mr. Warfield. Strike that.

Does the invocation of F transfer execution to 1-2-3 commands?

MR. GUTMAN: Objection.

THE COURT: He may answer.

A. You mean if I just was to type in F or if I was to type that F?

Q. Where does that F send you?

A. This F—and since we're on the record here, it's the F under worksheet global F, WGF. That sends us into the defaults format menu equivalent command.

Q. Are those Lotus menus or are they Quattro menus?

MR. GUTMAN: Objection. Are which Lotus—

MR. REBACK: All I'm trying—

[4-197] THE COURT: Overruled. You may answer the question.

A. Okay. Defaults format is just a menu that's used within our native menu tree.

Q. Okay. If the Key Reader—when the Key Reader gets to a format—an F keystroke, what is displayed on the screen, a Lotus menu or a Quattro native menu?

A. Our native menu from defaults format.

MR. REBACK: That's my point, your Honor.

THE COURT: Okay. I understand.

MR. REBACK: I just want to note for the record Plaintiff's Exhibit Number 3 is the redacted version—number 13 is the redacted version of the file Mr. Gutman has been talking about that was moved into evidence during Phase I by the plaintiffs.

THE COURT: All right.

MR. GUTMAN: We did not move that into evidence in Phase I, your Honor.

THE COURT: Is it not yet moved into evidence?

MR. GUTMAN: No, and there was an objection based on the redaction.

THE COURT: Well, all right. The record is what I'll depend on and what you better depend on so if there's something not in evidence that you want to get in evidence then it's up to you to do it before we end the **[4-198]** trial.

MR. GUTMAN: If I may ask one question based on his questioning.

THE COURT: All right.

RECROSS-EXAMINATION

BY MR. GUTMAN:

Q. The defaults format menu that Mr. Reback just asked you about is the same for Quattro Pro and 1-2-3, isn't that right?

A. They're—the same words appear.

Q. In the same order, correct?

A. From a macros perspective, yes.

Q. How about from the user's perspective looking at the screen? If you leave aside the fact that in 1-2-3 they appear horizontally and in Quattro Pro they appear vertically, isn't it the fact that it's the same words in the same order?

A. Leaving aside that fact, yes, that's true.

MR. REBACK: One last question.

REDIRECT EXAMINATION

BY MR. REBACK:

Q. Why is that?

A. Well, it's a rather fortuitous thing, actually. It turns out that the formats are numbered within the file since everything

to a computer is a number, and the [4-199] numbering that's used happens to be the same order as they're numbered in these menus. So the easiest and most efficient way for us to be able to tell what to put in the file when—

THE COURT: Well, now, you say it happens, but that happens by your choice, doesn't it?

THE WITNESS: I'm sorry?

THE COURT: It happens by your choice. It wasn't just an accident.

THE WITNESS: It happens by our choice to be able to read that point in a file.

Q. Who gave those numbers, you or Lotus?

A. The numbers were determined by Lotus.

Q. Is this an issue of file compatibility or macro compatibility?

A. It's an issue for file compatibility.

THE COURT: All right. We'll complete the side bar conference.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

Courtroom 11
Thursday, April 1, 1993
Boston, Massachusetts

LOTUS DEVELOPMENT CORPORATION

—vs.—

BORLAND INTERNATIONAL, INC.

NON-JURY TRIAL PROCEEDINGS
DAY 5

Before:

THE HONORABLE ROBERT E. KEETON
United States District Judge

APPEARANCES:

O'SULLIVAN, GRAEV & KARABELL, by Henry B. Gutman, Esquire, Kerry L. Konrad, Esquire, Joshua H. Epstein, Esquire, and Paul M.

O'Connor III, Esquire, 30 Rockefeller Plaza, New York, New York 10112, on behalf of the plaintiff.

WILSON, SONSINI, GOODRICH & ROSATI, by Gary L. Reback, Esquire, Peter N. Detkin, Esquire, Andrew G. Konstantaras, Esquire, and Isabella E. Fu, Esquire, Two Palo Alto Square, Palo Alto, California 94306

—and—

DONNELLY, CONROY & GELHAAR, by Peter E. Gelhaar, Esquire, 176 Federal Street, Boston, Massachusetts 02110, on behalf of the defendant.

TIMOTHY J. WILLETTE, RPR-CM
Official Court Reporter
U.S. District Court—P.O. Box 347
Boston, Massachusetts 02101-0347
(617) 248-0604

[5-2] (9:05 a.m.)

PROCEEDINGS

IN OPEN COURT

THE CLERK: Court is is now in session. You may be seated.

MR. GUTMAN: Good morning, your Honor.

THE COURT: Good morning.

MR. REBACK: Your Honor, I wish to apologize. Because of the rain our courier didn't show up and that's the reason we don't have—

THE COURT: All right.

MR. REBACK: There are a couple of housekeeping matters we could productively use the time for—

THE COURT: All right.

MR. REBACK: And once again let me say, sir, I apologize.

I'd like to begin by making the proffer that you had given me the opportunity to with respect to the 2.0, 2.01 issue. Yesterday I believe the Court said that the Court would give

me the opportunity to go back to the transcripts and make that proffer, and I'd like to do that now, sir.

THE COURT: All right.

MR. REBACK: First I'm going to explain why it **[5-3]** is I think that there is a macro compatibility problem between versions 2.0 and 2.01, then I'm going to call the Court's attention to the prior rulings in this case, and then as a result of that I'm going to move to strike the portion of the presentation yesterday which dealt with Version 2.01. If your Honor would just hear me out, I would appreciate that very much.

THE COURT: All right.

MR. REBACK: Okay. Now, in the *Paperback* case, I'm first going to cite the deposition of David P. Reid taken January 30th, 1990. On page 7 of that transcript Mr. Reid identifies himself as vice-president and chief scientist, spreadsheets, for Lotus. On page 61 he is examined and he is asked the following questions.

"Question: In the context of Release 2, do you know if there were issues of compatibility with respect to earlier versions of 1-2-3?

"Answer: Yes, I know there were many.

"Question: Did those issues present difficulties with respect to developing Release 2.0?"

There are objections and colloquies, and essentially no answer to the question, but the Court is—we're certainly willing to let the Court examine the transcript.

In the *Paperback* trial I'm now going to cite the **[5-4]** transcript from the eighth day of civil non-jury trial. I don't have a date on this, sir, but it's the eighth day, and I'm going to cite from page 156 of Volume 8. And the question is asked of Mr. Manzi, chairman of Lotus:

"Question: And the problem with Release 2.0 is it was not compatible with worksheet files that had been written with Lotus 1-2-3, Release 1A?

"Answer: No, I think it was a little bit more detailed than that. A lot of it had to do with undocumented features that were in 1A, actually 1.0 and 1A the people had used and we hadn't even represented for use or documented, and some of

our large accounts, for instance, needed those particular things supported. So there is this gradation of how compatible can you be or do you want to be while at the same time protecting the investment that's gone before and thinking prospectively about enhancing the product and the technology and so on, and so you encounter—

"Question: And so you encounter resistance"—"and so you encountered resistance from your customers because your customers perceived that Release 2.0 was not compatible with Release 1A?

"Answer: Not sufficiently compatible for some of our customers, yes, sir."

Now, that ends the quotation at that point. Now, we have previously filed in our briefs quotations from Mr. [5-5] Manzi in this case in his deposition taken in this case, particularly on August 22nd and August 23rd.

For example, August 22nd, page 173, there is a reference to a document which he's being shown, Version 2.0, a discussion of compatibility. Those are as set forth in our briefs and I'm not going to repeat them, but I would like to quote from page 174—oh, I beg your pardon. I can't quote from this because this portion is confidential. But let me just say that there's a discussion of the difference between Release 2.0 and 2.01, and I will hand that up at the appropriate time to the clerk.

Now, your Honor, in this case—

THE COURT: Well, why don't you just hand it to me right now and let me look at it?

MR. REBACK: Yes, sir.

(Document handed to the Court.)

(Pause—Court is reading.)

MR. REBACK: Okay, sir. Now let me continue, if I might—oh, I'm sorry.

THE COURT: Well, wait just a moment. I want to finish reading what you've highlighted.

MR. REBACK: Yes, sir. I'm sorry.

THE COURT: All right.

MR. REBACK: Now, I now would call the Court's attention to Docket Number 45 in this case, which is [5-6] Borland's

Memorandum of Law in Support of Defendant Borland's Motion to Compel Production of Lotus Patent Applications and Development Documents. That was filed I believe May 24th of 1991, and the second of the questions statement—the second of the questions stated there deals with the issue of whether we are entitled to the development documents with respect to versions of Lotus released after those with which we are charged with infringement in the complaint. And the question set forth on page 4 of the memorandum: Are these documents likely to constitute or lead to admissible evidence where Borland believes that such documents will show that 1-2-3's command menu tree is used as a system or process for maintaining macro compatibility? And there is extensive discussion in the memorandum starting at page 27 about the various releases and the reason we want development documents, and all of it goes to macro compatibility.

That motion was argued, your Honor, before this Court on June 18, 1991, and the way the record stood at that point, I had asked for development documents for, I believe, three or four of the subsequent releases, 1-2-3 G, 1-2-3 for Windows, 1-2-3 for the Macintosh and—

THE COURT: Well, when were those in relation to 2.01?

MR. REBACK: Afterwards, sir. They were after [5-7] 2.01.

THE COURT: Well, how are you arguing that discovery in relation to those later releases would have anything to do with the comparison between 2.0 and 2.01?

MR. REBACK: I had also at that time scheduled a number of depositions going both to those versions and to 2.0 and 2.01.

THE COURT: Well, now, wait a minute. The fact that you asked for things going back as well as things going forward and I disallowed going forward doesn't seem to me to make your point that you've been—you had already had an opportunity for discovery with respect to the releases as to which infringement was charged, hadn't you?

MR. REBACK: No, sir—oh, with respect to the releases with which infringement is charged, yes, sir.

THE COURT: All right. What were the releases with respect to which infringement was charged?

MR. REBACK: 1.0, 1A and 2.0. That's what's in the complaint. Please hear the rest of my argument.

THE COURT: All right.

MR. REBACK: I do understand your Honor's question and I will address it.

THE COURT: All right.

MR. REBACK: I had depositions pending going to a number of these things, and the ruling your Honor gave **[5-8]** me on my motion to compel was that I was not entitled to any discovery with respect to versions released after those with which we were charged in the complaint with infringement, and at that point we cancelled the depositions.

And I'm now going to read from the transcript or I will further point to the transcript where the Court made that ruling, and I'll start on page 26. And at the top of that page at line 2 I say: "Borland's motion to compel goes to a number of issues, including macro compatibility." And then there's colloquy with the Court, and on page—at line 17 of that page: "No, I don't think I should allow discovery for that purpose, because it goes—what you're arguing to me, I take it, is, you want to make discovery not with respect to the time period when Lotus 1-2-3 was being developed, those versions on which the present action is based, but you want to make discovery of later developments for the purpose of confirming something or other about Borland's motivation back in the earlier time." And I say, "Yes, sir."

Then the Court goes on at page 27, line 2. The the Court discusses the need to cut down on discovery, and then at line 12 addresses the depositions I have scheduled. And the Court says, quote: "I don't mean to be trying to decide that you can't take depositions of particular people, because I don't know what the subject—what subject matter **[5-9]** might come up in it, but it is my view at this point that if you are using these depositions for the purpose of doing the same thing you wanted to try to do by the document demands for the later period of development, I would expect to rule the

same way. You may be back here before me on that subject. I don't want to try to make an abstract ruling on it, but that's very likely what's going to happen. At some point, you know, I begin to think about whether Rule 11 is being violated by demands that disregard totally the rules with respect to not only Rule 11, but also other discovery rules and so forth," your Honor.

Now, I renewed—we cancelled the depositions. We obviously did not get the development documents. We got some documents with respect to Version 2.01 from Mr. Gutman, but we were never able to trace down the difference in the incompatibility, and so I raised it again on the renewed motion for summary judgment in this case. And I said I'm still being charged with something—that Lotus' argument is based on something with which I'm not charged with infringement.

And there was oral argument on that subject Tuesday, May 19th, 1992 in this court, and the Court is having a colloquy with Mr. Gutman. And the Court says at line 22, quote:

"What puzzles me is why you don't just ask me to **[5-10]** allow you to amend your complaint and claim copying of 2.01 as well.

"Mr. Gutman: That's my fallback, your Honor." I'm onto the next page, line 1.

Quote: "If we had to, obviously we would ask for leave to amend to conform to the proof.

"The Court: Well, why don't you? Is there some other collateral consequence that you're worrying about?

"Mr. Gutman: Absolutely not, your Honor." And there's further colloquy.

Subsequently Lotus made amendment in this case to bring in the Key Reader. There was an earlier stipulation with regard to similarities between 2.0 and 2.01, but that stipulation only went to issues in the case at that time, not to the Key Reader issues. Key Reader issues are issues of macro compatibility. They could have amended at any time. Certainly they could have amended at the time—

THE COURT: At any time after they knew about Key Reader?

MR. REBACK: Yes, sir. They could have—

THE COURT: Not before?

MR. REBACK: Well—

THE COURT: Well, they wouldn't have any reason to before they knew about Key Reader, would they?

MR. REBACK: I believe if they're going to [5-11] demonstrate—attempt to demonstrate that I am infringing Version 2.01, they should have charged me with that from the beginning.

THE COURT: Well—

MR. REBACK: But my only point here, your Honor, we're on the Key Reader Phase, and that's all I'm trying to argue here.

THE COURT: Now, let me understand. Are you now contending that infringement of 2.01 is an issue in this case?

MR. REBACK: I'm contending it is not an issue in this case.

THE COURT: All right. Then the fact that they could have amended to make it an issue is irrelevant. It's not an issue unless I'm told otherwise in a few minutes. 2.01 infringement is not an issue.

MR. REBACK: Very well, sir. That is my point.

THE COURT: Key Reader infringement is an issue.

MR. REBACK: Yesterday during the examination of Mr. Warfield, I attempted to ask him whether the Key Reader is compatible with Version 2.01. That answer was not permitted to be given. I will proffer for the record the answer is: It is—it is to be compatible with Version [5-12] 2.01. We don't know whether it's compatible with Version 2.0. It was never tested in that respect. It was never intended to be compatible—

THE COURT: Well, now bear in mind that the reason I made the ruling I did was your stipulation about the scope of his testimony, and I ruled the way I did because I told you you're fighting the stipulation.

MR. REBACK: Okay.

THE COURT: I wouldn't have ruled that way if I didn't have the stipulation before me.

MR. REBACK: Your Honor, I would like to discuss at some point—I understand your Honor's ruling to that effect. I would like to discuss at some point the circumstances under which that stipulation was signed, but I don't want to burden this particular argument with it.

My point is exactly the one made by your Honor. I'm not charged with infringement of Version 2.01. That was the only one demonstrated yesterday. I don't know what the differences are between Version 2.0 and 2.01, and the Court did not allow me discovery into that version because of the position Lotus took at that point.

THE COURT: Well, but I said when I made that ruling you can come back before me if you show me some reason it's material. At the time I made that ruling there was no claim of infringement before me either with respect [5-13] to 2.01 or Key Reader. Later Key Reader became an issue. It was up to you to come back and ask me for a change in my ruling on discovery if that change in the issues in this case justified it. You didn't, did you?

MR. REBACK: No, sir. I think it's their burden to demonstrate that what I'm infringing—

THE COURT: Listen, it is never their burden to ask me to come back and rule in your favor on something that I said I'm not ruling on now. If it ever becomes material, you may come back to me. You didn't.

MR. REBACK: Well, that is correct, your Honor.

THE COURT: That's no burden of theirs to come back and say, "Your Honor, we want you to change your ruling in our favor and now rule in their favor because of the change of circumstances." Of course that's not their burden.

MR. REBACK: Your Honor, at the time I made the original motion, your Honor raised the issue of Rule 11 sanctions, and quite frankly, sir, I was not going to push the issue in light of your Honor's view of it at that time.

THE COURT: That didn't stop you from coming to me to say—and you could have done it the way you've done in many other instances—"I don't want to be in violation of the Court's ruling, but now there is a change [5-14] of circumstance." Of course you're free to do that.

MR. REBACK: All right. Your Honor, I understand your Honor's point on that, and I will not argue with the Court. I will continue to press the point that as of right now, I am only charged with violation—with infringement of Version 2.0 and not with Version 2.01, and the only thing that was demonstrated yesterday was Version 2.01. That's my point, sir.

THE COURT: All right.

MR. GUTMAN: May I respond, your Honor? And since we traveled light today not expecting this issue, I'd like to borrow from my brother the stipulation he signed.

(Document handed to Mr. Gutman.)

MR. REBACK: And the cover letter as well. There you go.

MR. GUTMAN: Thank you very much.

I think Rule 11 is an appropriate subject this morning, and it pains me to raise this, but my brother, Mr. Reback, has flatly, flatly misrepresented the record in this case. And as I said, I didn't expect to be arguing this this morning, so I don't have all the quotes, but I'd be happy to bring them back after lunch if it would help.

The issue on the discovery motion to which he refers was not versions of 1-2-3 that came out after those named in the complaint. It was versions of 1-2-3 that came **[5-15]** out after Quattro was released. That's where we drew the line. We did not draw the line based on the three identified in the complaint.

2.01 came out before Quattro was released. We did not block any discovery with respect to 2.01. We produced thousands of pages to them, including pages that talked about this supposed compatibility issue. We gave them free rein in depositions on that issue.

Indeed, Mr. Reback's former colleagues took the deposition of Jill Lashway on August 7th, 1991. Lisa Olson and Mitchell Zimmerman of his old firm, Fenwich & West, took that deposition. Page 82:

"I believe you stated earlier that you were also the program manager for Version 2.01 of Lotus 1-2-3?"

"Yes.

"And can you tell me the reasons why Lotus decided to release Version 2.01?

"Answer: 2.01 was a maintenance release of 1-2-3." And it goes on.

They questioned her on precisely this subject, and she was the program manager. We never refused to produce anyone because they were relevant to 2.01 and not to some other release. Never happened. The depositions that were at issue—and I'm doing this from memory, but I think I'm pretty close—at the time of the motion Mr. Reback's **[5-16]** referring to were people involved in 1-2-3 for Windows, 1-2-3 G, perhaps 1-2-3 Release 3, all of which came out afterwards.

THE COURT: All right. Now, let me interrupt.

Mr. Reback, I'm being told by Mr. Gutman that the line that was being drawn and the issue that was before me in that discovery dispute was whether you should be allowed discovery for releases after Quattro came out, not for releases after the releases by Lotus as to which infringement had been explicitly charged in the complaint. Now, is that correct?

MR. REBACK: That was the subject matter of my motion, your Honor.

THE COURT: All right. Then it seems to me that by itself defeats the argument that you're making to me now. You had an opportunity to discover with respect to 2.1.

MR. REBACK: Well, your Honor, my point was that the ruling that your Honor made was not limited to—to that, but—

THE COURT: The ruling I made was limited to the issue before me presented by your motion, and things you read to me just a few moments ago indicated that if you had a different motion to present because of change of circumstances, I was not ruling on it and you could come **[5-17]** back to me.

MR. REBACK: Very well, your Honor. I understand your Honor's point.

THE COURT: You know, you really shouldn't have withheld that information from me in making your argument. You were telling me that I had forbidden discovery on 2.1.

MR. REBACK: Your Honor, in absolute good faith, it was my understanding when I came out of that argument that

I could not press that issue. But the only thing I'm pressing today, sir, is the issue that you stated before.

I understand your Honor's statement on discovery. The only issue I'm pressing is: Am I charged with infringement on the Key Reader Phase of Version 2.0 or 2.01?

I accept your Honor's ruling on discovery. Your Honor has ruled I could have taken that discovery—

THE COURT: Now, I understand the record to be as follows: That you are charged with infringement of releases up through 2.0—

MR. REBACK: Yes, sir.

THE COURT: —and with infringement of Key Reader.

MR. REBACK: Yes, sir.

THE COURT: Now, I do not understand the **[5-18]** record in this case to mean that Lotus is claiming infringement of 2.1 as distinguished from the earlier releases. Am I correct?

MR. GUTMAN: Your Honor, we are relying on the stipulation which says that for these purposes they are the same.

THE COURT: That 2.0 and 2.1 are the same?

MR. GUTMAN: Exactly.

THE COURT: Well, read me the stipulation.

MR. REBACK: Why don't you just hand it up.

MR. GUTMAN: Well—

MR. REBACK: Just hand it up. Why don't you—it's two paragraphs.

THE COURT: Well, all right.

MR. GUTMAN: Your Honor—

THE COURT: Read it and then hand it up. Read what you want to read out of it and then hand it up so I can see the whole thing.

MR. GUTMAN: Okay. "The menu commands in Lotus 1-2-3 Release 2.01 are the same as those in Lotus 1-2-3 Release 2.0, and are displayed to the user in an identical manner during operation of the program, and there is no difference in the user interfaces," with the S in parentheses, "of the two versions that would have any affect on the issues in this case."

[5-19] Now, the reason for that stipulation was a matter of convenience for both sides, your Honor, because both sides at

that point had come to understand that Release 2.0, because it was only in the market for a very short period of time, was no longer readily available to either of us. There was copy protection on it, so it was difficult to load on hard disks, it was difficult to use with our experts, it was difficult to use in depositions, and it was difficult to use for demonstrations. And each of us in reliance on this stipulation has consistently used Release 2.01 for purposes of our demonstrations.

Now they didn't show 1-2-3 yesterday in their demonstration, but if your Honor recalls the videotape demonstrations that they submitted on the summary judgment motion, they used 2.01 just as we did. Both sides have used it as being interchangeable with 2.0 because for all purposes relevant here—and macro compatibility, I should add, is not a new issue with the Key Reader. It was an issue before too. That was their reason they said that they were doing what they were doing before Key Reader. It's not a new issue to this case. It's been an issue in this case all along. And for all relevant purposes in this case, any effect on the issues in this case, we had stipulated that 2.0 and 2.01 are the same.

Now, they've infringed both, because what they **[5-20]** did is that—you know, it's like the second printing or second edition of a book. If the text is not different, then the infringement is the same. And if it would simplify things, your Honor, I would be quite happy to move under Rule 15(b) at this point to amend the pleadings to conform to the proof, because both of us have done it.

THE COURT: Well, I don't think I could allow you to do that for the reason that I really pressed you as to why you weren't doing that earlier and you weren't doing it, and so I don't think I ought to give you some advantage that might grow out of being allowed to make a trial amendment at this point for that purpose.

So I'11 consider the issue on the other basis. I just—

MR. GUTMAN: Well, your Honor, what—

THE COURT: Trial amendments are appropriate where an issue has been tried by consent. They are not appropriate where one side holds back from making the amendment ahead

of time so everything can be developed in relation to discovery and preparations for trial, and then wants to ask me in the midst of the trial to allow the trial amendment over opposition, and that's what you're doing. I will not do it.

MR. GUTMAN: Your Honor, the—I'm not looking for an unfair advantage. What I'm looking to avoid **[5-21]** is being unfairly sandbagged by my brother. When I put on a demonstration in reliance on a stipulation—

THE COURT: Well, I understand that argument—

MR. GUTMAN: —and—

THE COURT: —and so I consider it, but I consider it on the record of pleadings as they are, especially in light of the fact that I in effect pressed you as to why we don't simplify the procedural posture by getting the amendment earlier.

MR. GUTMAN: I don't have the complete transcript, your Honor, so I don't recall where that colloquy ended, but I would like to reserve the right to look at the rest of it—

THE COURT: You may.

Mr. Gutman: —and see, because I think there may well be an explanation of why we didn't do it at that time in that transcript. Among other things, we didn't want to run into the argument that it was too late in the game, and we were relying then as we are now on the stipulation. We did not believe it was necessary because we had a stipulation that for all relevant purposes they were the same.

And I believe we could also—and I don't want to quarrel with your Honor's ruling, but there is evidence **[5-22]** in the record that both sides throughout, and not just in their demonstrations, have treated 2.01 and 2.0 as interchangeable. If your Honor would look again at all the expert declarations that were put in by both sides and the witness testimony, both sides have in fact tried it as if it was the same.

Now, I don't need that and it doesn't matter to me if he doesn't get out of this stipulation and if it sticks, but the motion before your Honor by Mr. Reback this morning is to strike my entire demonstration which he conveniently raises today when the machines aren't here, Mr. Roshfeld isn't here. He declined to cross-examine Mr. Roshfeld yesterday. If he

cross-examined Mr. Roshfeld, he could have found out that the, quote, incompatibilities between 2.0 and 2.01 had absolutely nothing whatsoever to do with any of what was demonstrated yesterday, had nothing whatsoever to do with the menu commands. Indeed, if your Honor—perhaps I should make a proffer for the record here.

The incompatibility—apart from cleaning up bugs, which was the principal purpose of the maintenance release in 2.01, the incompatibility at issue there did not involve the menus, did not involve the menu commands, did not involve anything that Mr. Roshfeld showed yesterday. What it involved was the question of whether in doing an **[5-23]** average under-the-average function, which nobody has shown in this case, Lotus 1-2-3 treated a cell that was a label rather than a number as being a non-event or being a zero.

There was a bug in Release 1A and in Release 1 of Lotus 1-2-3 that caused those labels to be treated as zeros. So that if you created an average of a range of numbers and it included within the range a cell that was—a label rather than a number, your average would be off, because the total would be divided by one more number than it should have been.

Now, in preparing Release 2, Lotus discovered this and corrected it. Unbeknownst to Lotus at that time, a very big customer, one of the country's largest accounting firms, had found it first and developed lots of applications working around that bug. So when Lotus corrected the bug, it made all of that accounting firm's applications no longer valid. That was the macro compatibility inconsistency that was fixed in 2.01, and Mr. Roshfeld—I would represent to the Court that Mr. Roshfeld could so testify that it had nothing to do with his demonstration. He would have—if Mr. Reback had crossed him on this point yesterday, and I would—and if this is an issue where there's any chance that the Court is going to grant that motion to strike my demonstration, your Honor, then I would request leave to recall Mr. Roshfeld so that we can put this on the record **[5-24]** and resolve it once and for all.

But I think it's totally unfair when we've relied on the stipulation, as I said yesterday we were doing, and used the same product that all of us have used for demonstrations for purposes in this case for him to come in the next day when the machines are gone, the witness is gone, and say, "Doesn't count. Wrong problem. Strike the demo." Thank you.

THE COURT: All right. Let me see the stipulation.

(Document handed to the Court.)

(Pause - Court is reading.)

THE COURT: All right. You wanted to speak further, Mr. Reback?

MR. REBACK: I don't think so. I'm prepared to move on.

I will say as the record reflects from yesterday, we asked that Version 2.0 could be put on for the computer demo, and technically they were not able to do so yesterday.

THE COURT: Yes, but that has been explained to me.

MR. REBACK: Yes, sir.

THE COURT: All right. Now, your motion is denied.

First, I think you are fighting the stipulation. **[5-25]** It is true that the stipulation was not focused specifically on the issues in Key Reader because they were not in the case at that point, but the issues that are material to this demonstration were in the case at that point. And let me explain more fully my reason for coming to that conclusion and in the process also I am stating an additional ground quite independent of that for my denying your motion.

Let me start with the point that there's an analogy in my mind to the constantly recurring argument that a party is entitled because of ambiguity in a written agreement, a stipulation or anything else, to offer evidence and have that ambiguity resolved in favor of the party who is advancing the proposal.

The law is crystal clear, although all too often counsel forget about it in presenting arguments on the question, that you have to show more than ambiguity. You have to show ambiguity in a way that might be resolved in your favor on the basis of a proffer of expected evidence that might reasonably be anticipated.

Now, if there were differences between Release 2.0 and 2.01 that had a bearing on that demonstration that I was seeing yesterday, I could well understand if the demonstration crashed that 2.0 might have been different and it might not have crashed with 2.0. But there was no crashing in the demonstration. There were two or three **[5-26]** points at which the error messages came, but there were explanations in the course of the demonstration itself and the inferences I might draw from seeing that demonstration, plus the explanatory evidence that was given by both witnesses during the process of the demonstrations that I saw. You have not proffered to me any sensible reason, a reason that would appeal to my common sense as a fact finder without any expert advice from anybody, that somehow or other the demonstration would work with 2.01 but wouldn't work with 2.0, as distinguished from the possibility which does make sense, that the demonstration would have crashed with 2.01 when it wouldn't crash with 2.0.

Now, also, the common sense high probability that that is an accurate assessment of the situation is reinforced by the terms of the stipulation. And I'm quoting the stipulation: "The menu commands in Lotus 1-2-3 Release 2.01 are the same as those in Lotus 1-2-3 Release 2.0 and are displayed to the user in an identical manner during operation of the program."

Now, I take it as I read the cover letter and the attachment that the earlier proposed stipulation had stopped there and then the added paragraph—added statement after the semicolon was added. Is that correct?

MR. REBACK: I believe that is correct, sir.

THE COURT: All right. So let's—suppose we **[5-27]** stop there.

Now, that tells me that not only the menu commands are the same in the two, but because they are displayed to the user in an identical manner during operation, the structure, the tree, the order, the sequence, are all the same, the arrangement in steps, so that a particular letter or symbol would mean something at one step that would be different from what its mean-

ing would be at a different step, a point that I was also shown again in the demonstration where C has one meaning at one step and it has a different meaning at another step. Because the stipulation is that the menu commands are displayed to the user in an identical manner during operation, that tells me the stipulation is that they were displayed in the same step order, the same structure, the same sequence in 2.0 and 2.1. So I have not heard from you any reasonable suggestion of any possibility that the demonstration I saw yesterday would have come out differently if it had 2.0 on there instead of 2.1.

Now, in addition to all of that, isn't it correct that you were prepared at this trial to show me a demonstration using 2.01? That's what you had proffered and expected to do. You changed the demonstration yesterday and instead of putting your—your advance preparation on and used the one that they had put there, but you were not **[5-28]** proposing to me to be able in this trial to show me a demonstration of 2.0 and Key Reader, were you?

MR. REBACK: I was only going to demonstrate the Borland Quattro products, your Honor. That's all I ever intended to present.

THE COURT: All right. Then what I'm saying to you is, you had not proposed to me that you were going to demonstrate a 2.0 product—comparison. Is it also true that in Phase I of this trial you showed me some demonstration that used 2.01? Mr. Gutman has told me so. Is that correct?

MR. REBACK: The only time that I can recall that is in the summary judgment phase we submitted a videotape, and in that videotape we took a portion of Lotus' videotape in which they demo'd Version 2.01. So in that sense—I'm now trying to respond to your Honor—I do recall that.

THE COURT: Yes. All right.

MR. REBACK: But that was not proffered by the—

THE COURT: All right. Now, for all the reasons I've indicated and the one more point that I made to you yesterday, there is another analogy that is instructive to me here.

When I have a photograph offered in evidence that **[5-29]** was taken at a time after the moment that was critical and that

we're trying to reconstruct in the courtroom with the aid of oral testimony and visual exhibits as well, I'm not supposed to exclude a photograph because it was taken at a different season of the year if it is explained to me in a way that I as fact finder or the jury as fact finder can assess its value for the probative purpose by taking account of the evidence that is offered to explain the difference between the exhibit and the conditions at the time that is relevant, and why they do not invalidate the other probative information that appears on the exhibit.

Now, I said to you yesterday and gave you an opportunity to offer any evidence you want that explains why the differences are material instead of just saying, "Oh, there are differences, therefore it's improper."

It's the same point that I was making about ambiguity. Simply showing ambiguity doesn't open up the field for you to offer all kinds of evidence if the ambiguity is between two possibilities, neither of which helps you. It's irrelevant, immaterial.

Your motion—your proffer is before me. I have heard it. I have considered it. It does not persuade me I should change my ruling—the ruling allowing the demonstration subject to particular rulings I made about portions of it yesterday and the testimony—or rather, it **[5-30]** was not so much the demonstration, I guess, as some of the testimony that was offered by the witnesses that I excluded.

Those rulings stand.

MR. REBACK: I understand your Honor's ruling.

I have a couple of other housekeeping matters. I think you'll—

MR. GUTMAN: Actually, before we do, your Honor, if I may, and I am not in doing this attempting to reargue your Honor's ruling on the motion for leave to amend, but I now have before me the transcript in question. And since I value and have done my best to safeguard my credibility in this courtroom during the years I've been here, I would like for the record if the Court will bear with me just to read what immediately follows where Mr. Reback left off.

THE COURT: You may.

MR. GUTMAN: He quoted me in response to your Honor's question about the "Why don't you? Is there some other collateral consequence that you're worrying about?

"Mr. Gutman: Absolutely not, your Honor." I believe that's where Mr. Reback stopped reading. My answer continues:

"It's simply Borland screaming about being surprised by this at the last minute; frankly, their version of the story in which they say, 'Oh, no, we were copying **[5-31]** 2.01, not the identical 2.0.'

"The Court: Is it correct that if your claim included copying from 2.01, that would not have affected the scope of discovery at all in this case, the scope of the issues?

"Mr. Gutman: Not at all, your Honor. We have in this case and in *Paperback* at every instance given full discovery with respect to 2.01, because for our purposes 2.01 and 2.0 were the same. We never drew a distinction, never. And although they made such a claim in their brief, there is no support to that in the record, zero, and they have cited none. We produced the full discovery record concerning the development of 2.01 in the *Paperback* case, in the *Paperback* case, your Honor, and all of that discovery was provided to them here. They have never been blocked from asking questions about 2.01 here. It's never been an issue. 2.01 was a maintenance release that fixed some bugs and some undocumented features, none of which had anything whatsoever to do with the user interface as they have stipulated. The stipulation wasn't some gimmick, your Honor. It was the appropriate stipulation because it was true.

"The Court: I understand that and I've perhaps diverted you from your principal argument. I don't care to have you press it any further on this point"—"I don't **[5-32]** care to have you press it any further on this point."

And then we went back to what I believe was the summary judgment argument. Thank you, your Honor.

THE COURT: Mr. Reback?

MR. REBACK: Can I go on to the next one?

THE COURT: You may.

MR. REBACK: All right. Thank you, your Honor.

May I just have my copy of the stipulation back that I've handed up?

THE COURT: Yes, you may. Oh, no. Here it is. Hand that back to—

MR. REBACK: Thank you, your Honor. All right.

THE COURT: Your other document I'm handing you back is the one that had some protected materials in it.

MR. REBACK: Yes, sir. You've handed to me now actually both of these items that I handed up.

The second item has to do with the software library that is being prepared for the Court. The Court does not yet have that software library, because as I understand it, there were some difficulties in loading the software.

Now, your Honor will recall that the last time in the last phase of trial, Lotus moved into evidence a number

* * *

[5-136] Mr. Konrad: 201, your Honor, beginning at line 23. This is examination of Mr. Warfield by Mr. Detkin which I believe was beyond the scope of what was asked of him by me; but in any event, it is a series of questions asking Mr. Warfield to opine as to whether or not—well, as to the nature of the 1-2-3.MU file. Whether it's part of the program, whether it's part of the code, are they instructions to the computer and so forth, all of which seem intended to try to present an opinion of Mr. Warfield either with respect to the definitions in the Copyright Act or as an expert on computer science or some other purpose for which he was not qualified in this deposition.

THE COURT: Well, from which I don't receive expert testimony, also. Now, as I read this, he is using program in the same sense as code. I don't know whether he means object code or what, but code.

MR. DETKIN: I believe that is discussed later on in the deposition transcript, that Mr. Konrad in fact counter-designated. I'd like to make—

THE COURT: Well, how is it relevant to the present trial, any issue in the present trial?

MR. DETKIN: It's relevant to the issue of we're not—

THE COURT: Let me see. Am I not correct? I'm not trying anything about what's in code? You haven't **[5-137]** given me any evidence about what's in code, isn't that correct?

MR. DETKIN: Your Honor, we have given you what's within the files of the program.

THE COURT: I use the word carefully. You have not offered me any evidence about what's in quotes "code", unquote, in this trial, have you?

MR. DETKIN: I believe—

THE COURT: In whatever sense you may be using code, have you?

MR. DETKIN: That's correct, your Honor.

THE COURT: All right. Then why should I be receiving these answers about what's in code?

MR. DETKIN: Well, I believe it goes to a point discussed this morning that the appellate court may not have as clear an understanding of the difference between the code and files as your Honor does.

THE COURT: You know, that objection leaves me completely baffled. Of course the appellate court is going to have an understanding of what you're arguing to them if you make the argument clear enough to them they can understand it; and if you don't, that's your problem.

So how would my receiving—it would seem to me my receiving this as a part of the record here would confuse the appellate court if they didn't have some kind of an

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

Courtroom 11
Friday, April 2, 1993
Boston, Massachusetts

LOTUS DEVELOPMENT CORPORATION

—VS.—

BORLAND INTERNATIONAL, INC.

NON-JURY TRIAL PROCEEDINGS
DAY 6

Before:

THE HONORABLE ROBERT E. KEETON
United States District Judge

APPEARANCES:

O'SULLIVAN, GRAEV & KARABELL, by Henry B. Gutman, Esquire, Kerry L. Konrad, Esquire, Joshua H. Epstein, Esquire, and Paul M. O'Connor III, Esquire, 30 Rockefeller Plaza, New York, New York 10112, on behalf of the plaintiff.

WILSON, SONSINI, GOODRICH & ROSATI, by Gary L. Reback, Esquire, Peter N. Detkin, Esquire, Andrew G. Konstantaras, Esquire, and Isabella E. Fu, Esquire, Two Palo Alto Square, Palo Alto, California 94306

—and—

DONNELLY, CONROY & GELHAAR, by Peter E. Gelhaar, Esquire, 176 Federal Street, Boston, Massachusetts 02110, on behalf of the defendant.

* * *

[6-94] which is the utilitarian aspect of what's going on here. I mean, let's face it. Neither Borland in the products up to the July 31st opinion, or Computer Associates, or SuperCalc 5, or Microsoft, and Excel 3.0 and 4.0 uses these words that Lotus says it has a copyright in. They don't use those words because they're the best words or their order is the greatest order or they're just the right words. They use the words as a system or method of operation. So I would respectfully maintain, your Honor, that what Lotus' claim with respect to the Key Reader demonstrates is it that this case is not about words. Why else would they be complaining about a blank box on a screen with no words in it?

Now, let me make one other point, and I would respectfully go around counsel table and approach the bench a bit for this particular point.

I'm now going to hold up in my right hand for the purposes of the record a screen shot of Excel 2.0 already in evidence. This is the file menu from Lotus 1-2-3 with the extract commands in it, the retrieve menu on the screen, a direct copy of the Lotus menu. This is the version in my right hand that Jim Manzi said he had no problems with. This is the version that the Court blessed in its *Paperback* decision and in Borland I.

In my right hand I have a Quattro Pro for Windows screen shot with a box that is blank.

[6-95] THE COURT: For the record, you meant left hand in that—

MR. REBACK: I beg your pardon, your Honor. In my left hand I have a screen shot of Quattro Pro for Windows with a box that is blank. Your Honor, how can it be that under copyright law what is in my right hand, the Lotus commands as displayed in Excel is absolutely okay, but what is in my left hand, a blank box with no words, is illegal? How can that be?

Your Honor, I cite *Baker v. Selden*, *Altai*, *Sega*, the Copyright Office amicus brief, the law professors' amicus brief, I cite a hundred years of copyright law that cannot be the right result and respectfully argue to the Court that the Court consider that argument in its deliberations.

May I hand this up to the Court if the Court needs to see it further?

THE COURT: Well, I have seen it. Thank you.

MR. REBACK: Thank you.

I would also point out for the record, your Honor, Excel 2.0—actually it's—yes, Excel 2.0 that I held up to the Court I believe in my right hand and I demonstrated that—and the product is in evidence so your Honor does not have to take my word for it. Your Honor can look at it. I just want to point out that as with respect **[6-96]** to what Mr. Roshfeld demonstrated yesterday in Quattro Pro, he demonstrated that in Quattro Pro's blank box you can cursor down from one command to the next. The commands in Lotus—in the Lotus file in Excel can be accessed by a keystroke, the same keystroke that's the first letter of the Lotus command, and your Honor can demonstrate that to the Court's satisfaction in chambers with the product that's loaded on the system.

One moment, if I may, your Honor.

(Defense counsel confer.)

MR. REBACK: I appreciate the Court's consideration, your Honor. Thank you very much for hearing me today.

PLAINTIFF'S FURTHER CLOSING ARGUMENT

MR. GUTMAN: Your Honor, if I may, I just have a few points I'd like to make.

Starting where Mr. Reback ended, this version of Excel, Excel 2.0, or I believe it was actually 2.1, that was before the

Court in the *Paperback* case and that is in evidence here, I'm not sure I understood my brother correctly, but if what he was saying was that you could use 1-2-3's keystrokes and cause the program to respond the way Mr. Roshfeld did with Quattro Pro for Windows using the Key Reader mode the other day, that's false, and I would invite the Court to try it.

[6-97] The piece of Excel, that version of Excel that Mr. Reback is showing your Honor—it was in his left hand and now it's in mine—is a help facility that allowed Excel users who were former 1-2-3 users and knew the 1-2-3 commands to easily discover what the Excel equivalents would be, because the Excel menus were themselves quite different. So what happened in this help facility—and I know we've been over this back in *Paperback*, but it's been some time—was, you could specify 1-2-3 keystrokes that would cause a certain thing to happen in 123 and what would come up in this box would be the Excel equivalent. So that if it took file exit to do the same thing as quit in 1-2-3, you could gain that information. It was simply a help facility. It was not in fact an alternate user interface and it did not in fact accept the keystrokes and do anything beyond telling you what the Excel counterpart would do.

Now, back—pardon me, your Honor. I'm losing my voice. Back to the beginning. The section of Philippe Kahn's transcript that Mr. Reback was relying on saying that this demonstrates that he customarily relies on Mr. Kohn for all these decisions does not fill the gap. That's exactly—I'm not going to read it, but it's exactly the section that Mr. Konrad objected to yesterday anticipating that they would argue this, anticipating that they would try to use it to bridge the evidentiary gap that has not and

* * *

[6-104] that, we certainly would have known it was there earlier than we did and would have dealt with it, but they never did. They were hoping they could do it by indirection.

Final point, your Honor. We are not talking about screen displays. We have never been talking about screen displays. We are talking about elements—Mr. Reback keeps trying to

talk about screen displays, but we never have. That's not what this case is about. That's not even what *Paperback* was about. Your Honor said so in the *Paperback* decision. What this case is about is copyrightable expression in the form of menu commands, a menu command hierarchy and structure, keystroke sequences and a macro language. They copied it before, they are copying it still. The only difference—two differences. One, they have abbreviated the commands down to initials, and frankly, I would say they are infringing whatever symbolic token, using an old familiar term, they choose. And two, they have hidden it from use. And I would respectfully submit that when you take our expression, neither of those is a material difference or anything that should matter as a matter of law.

Before I sit, the last thing I wanted to do, your Honor, I don't know whether your Honor has received the March issue of the Harvard Law Review yet. I'm sure your Honor will.

[6-105] THE COURT: Yes, it came through my front door slot a few days ago.

MR. GUTMAN: I had mine sent up just in case. Your Honor, there is an article in there by Professor Miller on the subjects that we have been litigating over here. We would respectfully submit that there is much that is useful and wise in that article and we would commend it to your Honor. Thank you.

DEFENDANT'S FURTHER CLOSING ARGUMENT

MR. REBACK: Let me begin with the last point first, which—I have not seen the final of Professor Miller's article, but I would note—first of all, of course the Court can read and consider whatever law review is appropriate. Professor Miller is in fact a consultant to Lotus and I would ask the Court to consider that when reading the article.

A couple of very quick points. I think Mr. Gutman confused two arguments that I made with respect to Mr. Roshfeld's demonstration and I just want this point to be very clear.

The menus that come up in the Key Reader are Quattro Pro native menus and your Honor can demonstrate that to the

Court's satisfaction in chambers with the product, and that was the point I was making about the menus.

Finally the last point with respect to this **[6-106]** Microsoft Excel display of Lotus menus. You can in this version that Mr. Manzi said they had no problem with, you can type in the keystroke that signifies the command and the cursor will move down to that command and you can then open up the subcommand. If I heard Mr. Gutman correctly, that's okay, as I understand it. What's not okay about the Borland product is that it executes the command, and I would respectfully suggest that is the distinction we're trying to make in this case. That is the very distinction that is at issue here. I would also point out that in the versions currently on the market, including those explained in Dr. Liddle's declaration, Versions 3.0 and 4.0, the commands will actually execute. But for the purpose of this argument here today, if I've made no other point, I want to make the point that it is the execution that Lotus cares about and not the words. Thank you, sir.

THE COURT: All right. Now, I believe we had an understanding, when was it, yesterday, about anything else that is to be done which concerns getting the record before me.

MR. REBACK: Yes, sir. We're going to take this away and as rapidly as we can get it back to the Court, in more than two weeks. Thank you, sir.

THE COURT: Thank you. We'll be in recess.

MR. GUTMAN: Thank you, your Honor.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF TIMOTHY A. BOYD

I, TIMOTHY A. BOYD, hereby declare under penalty of perjury as follows:

Experience and Qualifications

1. I am a Vice President of Burke Marketing Research, a nation-wide organization which provides marketing research services to hundreds of corporations in the United States. I work in the San Francisco Client Service Office of Burke Marketing Research.

2. Burke Marketing Research has been engaged in market research for 60 years, and is one of the largest and most respected market research organizations in the country. Each year Burke conducts more than 600 research studies, and surveys approximately 75,000 individuals annually. Burke Marketing is a member of the Council of American Survey Research Associations.

* * *

9. Based on my experience as an expert in the fields of market research and usage surveying and based on the user survey conducted in January 1991 under my supervision, I have reached the following conclusions regarding the use of Quattro and Quattro Pro:

- Almost 9 out of every 10 Quattro and Quattro Pro users (88%) do not use the 1-2-3 compatible alternative menu arrangement to any extent whatsoever.
- Most usage of the 1-2-3 alternative menus appears to be driven by the desire to run macros created using Lotus 1-2-3. Of the relatively small number of Quattro and Quattro Pro users who have ever used the 1-2-3 alternative menus to *any* extent whatsoever, fully 35% report that the *only* time they use the 1-2-3 alternative menus is when *someone else* gives them a worksheet with 1-2-3 macros. Furthermore, two-thirds of those who have ever used the 1-2-3 alternative menus agreed with the statement that they use the 1-2-3 alternative menu arrangements "because [they] want to run worksheets that contain macros created using Lotus 1-2-3."

10. In the remainder of this Declaration, I will describe in detail the manner in which the survey was performed.

* * *

34. Questions Q.7 asked the 60 respondents who stated that they had *ever* used the 1-2-3 menus which menu arrangement they were *currently* using. Twenty-nine (12% of the base of Quattro/Quattro Pro users) stated that they were now using the 1-2-3 alternative menu arrangement. In response to Questions 7 and 8, 21 respondents—8.7% of those who have ever used Quattro or Quattro Pro—stated that they use the 1-2-3 alternative menu arrangements most often.

35. The remaining questions (Q.9 through Q.10C) explored the users' reasons for using the Lotus 1-2-3 alternative menu arrangements of Quattro/Quattro Pro. In response to Question 9, two-thirds (66.7%) of those who had ever used the 1-2-3 menu arrangement agreed that they did so because they wanted to run worksheets that contained macros created using the Lotus 1-2-3 program. In response to Question 10A, 35.0% of those who had ever used the 1-2-3 menu arrangement agreed that they did so *only* when someone else gave them a worksheet with 1-2-3 macros.

36. In my professional judgment, the foregoing conclusions are accurate and reliable.

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above, except as to those matters that are stated on information and belief. As to those matters, I believe them to be true. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury this 26th day of September 1991 at San Ramon, California.

/s/ TIMOTHY A. BOYD
Timothy A. Boyd

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF EUGENE K. BUECHELE

1. I am currently Vice President, Engineering for the Network Systems Division of 3-Com Corporation. I make this declaration in opposition to the motion of Plaintiff Lotus Development Corporation ("Lotus") for summary judgment in the above-captioned matter. I have personal knowledge of the matters set forth below and could testify competently thereto if called upon to do so.

2. On September 15, 1980 I joined Personal Software, Inc. (later "VisiCorp"), a manufacturer and distributor of applications software for personal computers located in Santa Clara County, California. At the time, VisiCorp distributed and marketed the VisiCalc computer spreadsheet product under an exclusive license from Software Arts, Inc. Shortly after I joined VisiCorp, I became Director of Engineering with responsibility, among other things, for ongoing development of VisiCalc and of VisiCorp's VisiTrend/Plot computer software product.

3. VisiTrend/Plot had been designed by Mitch Kapor and licensed to VisiCorp. In October 1981, VisiCorp acquired all of Mr. Kapor's right, title and interest in the product for approximately $800,000.

4. I am familiar with the user interfaces of Lotus 1-2-3 Version 1A and VisiTrend/Plot and have compared them for purposes of preparing this declaration. Apart from some of the words in the menu command hierarchy, the control mechanism of the 1-2-3 user interface, including the two-line moving cursor, was present in and adapted from the VisiTrend/Plot product.

5. These control mechanism similarities are set forth in the following chart:

Feature	*Lotus v. 1A*	*VisiTrend/ Plot v. 1.0*
Full Word Command Names	Yes	Yes
Highlighted, Moving Cursor	Yes	Yes
Command Selection Based on Highlighting, with Command Activation based on subsequent <Enter> key	Yes	Yes
Command Activation based on Abbreviated Command Name	Yes: 1 Letter	Yes: 2 Letters
Ordering of Commands by Anticipated Frequency of Use	Yes	Yes
Command Long Prompts following Highlighting on Status Line	Yes	Yes
Switch from GRAPH Display Mode to GRAPH Command Mode	<Escape Key>	<Escape Key>

6. The text of the command menus of the core functions in VisiTrend/Plot, a graphics and plotting product, is also similar to that of the graphics or "2" function of 1-2-3.

7. Unlike VisiCalc, the VisiTrend/Plot product was never transferred to Software Arts as part of the litigation settlement between VisiCorp and Software Arts. Hence, Lotus never acquired ownership of VisiTrend/Plot when it subsequently acquired Software Arts.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct and that this declaration is executed this 27 day of September, 1991 at Palo Alto, Santa Clara County, California.

/s/ EUGENE K. BUECHELE
Eugene K. Buechele

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

SECOND DECLARATION OF EUGENE K. BUECHELE

1. I am the Managing Partner of the Prosper Group, a computer consulting firm. I make this declaration in support of the motion for reconsideration of Defendant Borland International, Inc. in the above-captioned matter. I have personal knowledge of the matters set forth below and could testify competently thereto if called upon to do so.

2. I have worked in the computer industry for 25 years with positions in engineering management, general management, and software development. During the period starting in September 1980 and ending in September 1984, I worked for Personal Software Inc. (later called VisiCorp) as the Director of Quality Assurance and Director of Engineering. During this period my responsibilities included managing and directing software developers; reviewing, selecting, and acquiring software programs from third parties.

3. On September 27, 1991, I executed a declaration in the above-captioned matter, which was subsequently filed as docket no. 135.

4. On August 12, 1992, I was shown this Court's July 31, 1992 order granting partial summary judgment for Lotus that stated that the holding in *Ashton Tate v. Ross*, 916 F.2d 516 (9th Cir. 1990), aff'g 728 F. Supp. 597 (N.D. Cal. 1989), was that "a document bearing 'a list of labels for user commands, many of which are common commands that were already available on other software programs' was not innovative or novel." This Court then went on to distinguish *Ashton Tate Corp. v. Ross* by stating, "[t]he relevance of that conclusion to this case is in some doubt in view of the fact that Lotus 1-2-3 is one of the programs on which the commands were already available. There is no evidence in this case that the commands available in 1-2-3 were common commands at the time of Lotus' authorship."

5. On that day I began reviewing other software packages that were available at the time of the initial release of Lotus 1-2-3. For commands that were new to 1-2-3 Release 2.0, I expanded my search to software that was available before the initial release of that version.

6. In the ten days since I began my analysis, I have been able to review the following 15 software packages, all of which were available prior to the relevant release of 1-2-3: VisiTREND/VisiPLOT, Personal Software, Inc.; VisiCALC, Personal Software, Inc.; VisiFILE, Personal Software, Inc.; PC-DOS Version 1.1, International Business Machines Corp. and Microsoft Corp.; MULTIPLAN, International Business Machines Corp.; MVS/TSO, International Business Machines Corp.; VM/CMS, International Business Machines Corp.; Fortran IV, International Business Machines Corp.; TROLL/1, National Bureau of Economic Research, Inc.; BBL II (Basic Business Language), Core & Code, Inc.; CP/M, Digital Research, Inc; Desktop/Plan, Marton, Inc.; SUPERCALC,

Sorcim; Cobol 68 and 74, United States of America Standards Institute; and dBase, Ashton Tate.

7. Approximately 80% of the 1-2-3 commands can be found in these fifteen earlier software packages, which use the same basic words as Lotus 1-2-3 to perform analogous functions. These commands are: Worksheet, Global, Format, Column-Width, Recalculation, Columnwise, Rowwise, Automatic, Manual, Iteration, Protection, Enable, Disable, Default, Printer, Left, Right, Top, Bottom, Page Length, Wait, Setup, Name, Quit, Directory, Status, Update, Other, Currency, Date, Time, Quit, Help, Clock, Quit, Insert, Column, Row, Delete, Column, Row, Column-Width, Set, Reset, Hide, Display, Erase, Titles, Both, Horizontal, Vertical, Clear, Window, Horizontal, Vertical, Syncy, Unsync, Clear, Status, Quit, Range, Format, Label, Erase, Name, Create, Delete, Labels, Right, Down, Left, Up, Reset, Table, Justify, Protect, Unprotect, Input, Value, Copy, Move, File, Save, Copy, Add, Substract, Values, Erase, List, Text, Numbers, Directory, Print, File, Graph, Type, X, Reset, Graph, X, Quit, View, Save, Options, Legend, Format, Graph, Quit, Titles, First, Second, X-Axis, Y-Axis, Grid, Automatic, Manual, Format, Indicator, Quit, Automatic, Manual, Format, Indicator, Quit, Skip, Color, Quit, Quit, Name, Use, Create, Delete, Reset, Quit, Data, Fill, Table, Reset, Sort, Data-Range, Reset, Go, Quit, Query, Input, Output, Find, Extract, Delete, Reset, Quit, Multiply, Regression, Output Range, Reset, Go, Quit, Create, Edit, Input-Column, Output-Range, Reset, Go, Quit, System, and Quit.

8. I have also compared the commands in a pre-Lotus 1-2-3 release of VisiCalc with the Lotus 1-2-3 command set and I have found that there are many similar commands.

9. Based upon this analysis, it is clear to me that the Lotus 1-2-3 command set is comprised of commands that were in common use in other copyrighted software programs before the release of Lotus 1-2-3.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct and this

declaration is executed this 27 day of August, 1991 at Palo Alto, Santa Clara County, California.

/s/ EUGENE K. BUECHELE
Eugene K. Buechele

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION
NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

DECLARATION OF JENNIFER DOS REMEDIOS

I, Jennifer dos Remedios, declare:

1. I am a legal assistant employed by the firm of O'Sullivan Graev & Karabell, counsel for plaintiff herein.

2. I am familiar with the structure of the menu commands of Lotus 1-2-3 in Quattro, the 123.MU menu tree file in Quattro Pro and VP-Planner Plus.

3. Submitted herewith as Exhibit A are two bound volumes containing prints of screen displays from Lotus 1-2-3, Release 2.01; the 123.RSC menu tree file in Quattro Version 1.01; the 123.MU menu tree file in Quattro Pro Version 1.0; and VP-Planner Plus.

4. I created the prints initially from the respective software programs while in operation, using a Compaq 386s/20 notebook computer connected to a Hewlett-Packard PrintJet XL color printer and specialized software called

Pizzazz Plus.

5. I combined the sets of prints of screen displays from Quattro and Quattro Pro to allow for their presentation together on a single page. All prints were then reduced to 78% in a process of color duplication. The legends were supplied by me and added in the duplicating process.

6. The contents of the volumes are as follows:

Volume I

Tab A—VP-Planner Plus

Each item in the top-level menu is shown with the cursor positioned on it.

The"Worksheet/Global", "Worksheet/Global/Format", and "Worksheet/Global/Format/Fixed" menu items are shown with the cursor positioned on each.

The "Range", "Range/Format", and "Range/Format/Fixed" menu items are shown.

The top-level menu is shown after the menu display was reconfigured according to the 1-2-3 display as in the videotape demonstration of Larry Roshfeld.

Tab B—Quattro Pro 1.0 Mouse/No Mouse

The screen display of Quattro Pro 1.0 is shown with the mouse device attached and not attached to the computer, and with the cursor on the "Worksheet" item in the main menu and on the "Worksheet/Global" item in the following sub-menus.

Tab C—Main Menu Commands

The screen displays for each item in the main menu of Lotus 1-2-3 is shown on the right-hand page, with screen displays for corresponding

items in the Quattro Pro 123.MU menu tree and Quattro 123.RSC menu tree shown on the facing pages.

Tab D—Worksheet, Range, Copy, Move, File, Print, Graph, Data, System Quit Commands

Screen displays for each item in the first menu following the selection of each item from the main menu in 1-2-3, together with the corresponding items in Quattro Pro and Quattro, are shown.

Tab E—Worksheet Sub-Commands

7. Items in the various sub-menus branching from the "Worksheet Commands" are shown for each program.

8. There are four commands in the Lotus 1-2-3 items depicted for which there are no Quattro counterparts. There are no such instances with respect to Quattro Pro. The total number of 1-2-3 commands depicted is 327. All 1-2-3 menus are shown to the fourth level.

Volume II

Tab A—Range Sub-Commands

Tab B—File Sub-Commands

Tab C—Print Sub-Commands

Tab D—Graph Sub-Commands

Tab E—Data Sub-Commands

I declare under penalty of perjury that the foregoing is true and correct.

Executed at New York, New York on November 1, 1991.

/s/ JENNIFER DOS REMEDIOS
Jennifer dos Remedios

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF JAMES C. EMERY

I, James C. Emery, declare:

1. I am Professor of Decision Sciences at the Wharton School of the University of Pennsylvania. A true copy of my current curriculum vitae is annexed hereto as Exhibit A. I make this declaration in support of the motion for summary judgment of plaintiff Lotus Development Corporation ("Lotus"), and in opposition to the cross-motion for summary judgment of defendant Borland International, Inc. ("Borland"). As the Court will recall, I also testified at Lotus' request in *Lotus Development Corporation v. Paperback Software International*, Civil Action No. 87-0076-K.

2. I consider myself professionally knowledgeable in the development and deployment of management information systems, computerized information systems, decision support

systems, and the use of such systems as critical resources in the efficient management of corporations at all levels from logistical planning to strategic decision making.

3. "Management information systems", or "MIS", is a discipline comprehending both transaction processing systems and "decision support systems", which are computer systems that provide data retrieval and modeling functions to aid human decision making.

4. I have taught both undergraduate and postgraduate courses in the areas of decision and information science, management, and computers at the University of Pennsylvania from 1965 to 1974 and from 1980 to present, and at the Massachusetts Institute of Technology from 1959 to 1965, except for leave periods. Among the courses I have taught have been courses in MIS, systems analysis (a management school course in the design and implementation of computer-based systems), and quantitative methods in business analysis. In my courses, I have regularly addressed user interface issues; for example, I have covered such issues in my systems analysis course, using Ben Shneiderman's book, *Designing the User Interface*, as one of the textbooks for the course.

5. During the 1990-1991 academic year, I served as the Grace Murray Hopper Visiting Professor in Computer Science at the Naval Postgraduate School in Monterey, California. I taught a course on information systems policy for organizations and engaged in research for the Director of Defense Information, U.S. Department of Defense, concerning the use of the Ada programming language for developing management information systems.

6. I agreed to consult for Lotus in this case last April, when I was in Monterey.

7. I have published two books and fifty-eight articles in the areas of management information systems, the economics of information, planning and control systems, logistical systems, computer networks, and decision support systems.

8. Since January 1989 I have served as Senior Editor of the *MIS Quarterly*, a leading academic journal in the information science field, with responsibility for the review of articles for publication. Part of my responsibility is to assign articles submitted to the publication to various of its approximately 25 associate editors for review.

9. From 1974 to 1979, I was Executive Director of the Planning Council on Computing in Education and Research for EDUCOM, a membership association of colleges and universities established for the purpose of improving the effectiveness of computing in higher education on matters pertaining to education, research, and administration. I also served as its President from 1976 to 1979.

10. While with EDUCOM, I had extensive contact with vendors of commercial software, including microcomputer software, concerning issues involved in the technical support of software products, including human engineering issues such as the importance of user interface design in making software easier to use and learn, site licensing, and product documentation.

11. I have served as a member of the boards of directors of the following corporations during the periods specified: (a) from 1979 to 1988, Gartner Group, a leading market research and technology assessment company in the area of information systems; (b) from 1973 to 1988, Data Architects, Inc., a company that develops custom software for large mainframe computers as well as large telecommunications systems for the financial services industries; (c) from 1978 to the present, Human Design Systems, Inc., a company which develops and manufactures computer hardware (primarily terminals); (d) from 1983 to 1986, Conceptual Instruments, Inc., a company that developed a desktop management software product for microcomputers; (e) from 1984 to 1985, Desktop Software Corporation, a company that developed database software for microcomputers; (f) from 1987 to the present, Colonial Management Group, a mutual fund company that manages a variety of mutual funds, for which I monitored the

development of a computerized system for the purposes of keeping track of shareholder information, permitting shareholders to switch investments between associated funds, and other similar functions; and (g) from 1989 to 1990, PSI, Inc., a company that developed customized large-scale application software.

12. In my capacity as a director of these companies, I have had considerable exposure to issues concerning the development, marketing, pricing, and licensing of software. As a director of Conceptual Instruments, Inc. and Desktop Software Corporation, I dealt with issues specifically concerned with the design and marketing of microcomputer software.

13. I have personally developed a variety of computer programs. For example, I developed a microcomputer-based program used by The Wharton School for the purposes of allocating and scheduling rooms for classes and other activities. In the past, I completed a feasibility study and the preliminary design for a computer software system for billing and receiving payments for pension fund contributions for the Presbyterian Pension Fund, a pension fund associated with the United Presbyterian Church. In addition, between 1956 and 1959, as an employee of Westinghouse, I participated in the development of several large-scale applications in the area of inventory control and production scheduling.

14. I have also consulted in the areas of logistics, MIS planning, and decision support systems. Some of these engagements involved the development of application programs. For example, in one such engagement I developed a microcomputer-based model for estimating construction costs for a large engineering design company. In a brief engagement for IBM Corporation, I helped to analyze the business potential of the "systems integration" market, in which IBM would provide part or all of a client's MIS support services under contract. In an ongoing research project funded by Digital Equipment Corporation, I am analyzing the use of information technology for the more effective planning and control of the company's business activities.

15. Prior and subsequent to my retention in this matter, I have communicated with John Seamster of Borland, whom I understand to be responsible for its educational market activities, as a user of its "Paradox" database software program. When Mr. Seamster invited me to attend a training session concerning Paradox after I agreed to serve as a consultant for Lotus, I informed him of the potentially adverse relationship I might have with Borland. Mr. Seamster later told me that he did not think my retention by Lotus would cause a problem with respect to the training session, and in addition he said that Borland's lawyers had asked him to invite me to talk with them. I declined that invitation, which I considered inappropriate.

The user interface of a computer software product

16. The "user interface" of a computer software product is the communicative link between the human user and the software product. The user interface defines the software product's appearance as well as its impression, communicative impact, and effect upon the user.

17. The user interface creates the product's identity by governing how users think about the product and how they conceptualize the tasks they deal with when using the product. The user interface is the principal determinant of the software's ease of learning and ease of use, and of the user's general satisfaction with the product.

18. Where a microcomputer application user interface employs menus, the menus and menu structure are typically communicative in nature, expressing the capabilities of the software to the user.

19. There is, for all practical purposes, potentially infinite variation in the expression of a product's menus and menu structure, through the selection of the set of tasks represented by the menu options, the choice of words or symbols used to represent the menu options, any associated description of menu options, and the clustering and sequencing of menu

options in a meaningful and comprehensible hierarchical structure.

20. The communicative and expressive aspects of the menus and menu structure are critical to the value of the software.

21. Moreover, for any given set of functions performed by a program, a practically infinite variety of user interfaces can be developed. Thus, it is technically unnecessary for software programs to employ similar user interfaces (or similar menus or menu structures) in order to provide comparable functionality.

22. In fact, microcomputer programs intended to address the same basic market—such as spreadsheet manipulation and calculation—need not provide the same precise range of functionality. There is almost always a large variety of ways to implement any individual operation or component of the program's functionality.

23. Designers must make judgments about the applications for which a software product will be used and then choose the "primitives" to include in the product that provide the best balance between ease of use, ease of learning, and cost. (A "primitive" is an individual operation the software is to perform without having to combine other operations.)

24. Different designers are likely to select for inclusion quite different sets of functions. Too much functionality clutters a program with features that are used rarely, makes the program difficult to learn, and adds to its cost. Too little functionality leaves the user dissatisfied because the program will not perform the tasks the user wishes or requires the user to combine multiple primitive steps to accomplish a desired task.

25. Typically, a small proportion of the commands in a software product will constitute the majority of its usage—that is, most of the tasks that users will wish to perform are articulated in a relatively few command sequences. A developer may find it sufficiently valuable from a marketing perspective to invest the resources necessary to provide

functionality that it does not anticipate will be frequently used. Nevertheless, the value of the product will depend primarily upon the small set of most frequently used commands and their associated functionality.

26. One basic distinction in user interfaces for microcomputer applications software is between those that are "command-driven" (*i.e.*, in which the software's operations are controlled by command sequences inputted by the user without the presentation of alternatives by the software to the user), and those that are "menu-driven" (*i.e.*, in which the operations are controlled by the user's selection of command options from developer-determined groups of choices presented by the software).

27. For "menu-driven" user interfaces, the organization of the menus into a menu structure (often depicted as a "menu tree", to reflect the branching of the structure and the logically dependent nature of those commands that "branch" from others) itself presents a major conceptual challenge to the developer. Many subjective decisions are required, concerning such matters as definition of the "primitives" to express in the menu structure, selection of the desirable number of levels in which the command options should be presented, selection of the command choices to group within each level, and selection of the words or other symbols to use to represent each command option.

28. The subjective decisions made in creating a user interface are among the most creative and challenging in microcomputer software development, and the most fraught with risky assumptions.

29. The assumptions reflect the developers' effort to project themselves into the minds of the intended users of the software and to imagine the approach to the problem addressed by the application that will satisfy the users' needs. It is very easy for developers to introduce elements that impede, rather than enhance, users' understanding.

30. The assumptions are risky because of the inevitable uncertainty as to how successfully the developers' approach will, in fact, meet the needs of users and eventually gain acceptance in the marketplace, as well as the possibility that other developers may simultaneously be developing products with similar functionality and/or different interfaces that users might prefer.

31. As Prof. Olson points out in her article, "The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS" (Judith R. Olson and Gary M. Olson, *Human-Computer Interaction*, 1990, Vol. 5, pp. 221-265), such cognitive principles as may be emerging in user interface design do not shed light as to *why* people buy software. There is as yet no science to predict what makes for a successful software product.

User interface design in software development

32. I have read the declarations of Prof. David Kieras, Dr. David Liddle, and Prof. Judith Olson submitted by Borland in this action, and the declarations of Prof. Bernard Galler and Mr. Mitchell Kapor submitted by Lotus, to which they largely respond. In criticizing Prof. Galler's statements concerning the relative significance of user interface design in the development of a successful software product, both Prof. Kieras (at ¶¶ 43-49) and Dr. Liddle (at ¶¶ 38-40) observe broadly that user interfaces for software programs of widely divergent types may be more or less interactive or complex, depending in part on their nature or intended purpose.

33. The degree of interactivity or complexity to employ in the user interface of a software product is itself a design decision made by the developer. Whenever humans are involved—which is almost always the case except in a completely automated process—user interface design issues must be addressed. (Even for a completely automated process, a human interface is needed for a human to set parameters and rules to guide the operation of the computer program.)

34. For example, it is not apparent to me why the example Dr. Liddle gives (at ¶ 40), of a "program to control the flow of traffic", would not involve complex user interface issues if it was intended for the support of human decision-making. The success of such a product may depend heavily on the effectiveness of the interface (*e.g.*, a graphical display of the road network with icons to show congestion points).

35. Moreover, whatever validity there may be at the most general level in the proposition that Prof. Kieras and Dr. Liddle advance, it has little to do with the software involved here.

36. I am unaware of any successful microcomputer software application in the marketplace for which the user interface design is *not* a major component of the product. Indeed, Prof. Kieras in his declaration (at ¶ 47) uses a "word processor"—by which I understand him to mean a microcomputer word processing application product—as his example of a product requiring considerable effort in its user interface design.

37. Almost all software products raise user interface design issues, and the design of a good user interface is almost always relatively expensive, regardless of the computational platform on which the software is intended to operate, *e.g.*, mainframes, microcomputers, etc. Traditionally, mainframe software user interfaces have not been particularly "friendly" because the economics of the mainframe market generally favor imposing a relatively heavy burden of training on the user instead of expending the resources necessary to develop user interfaces that are easy to use and easy to learn.

38. The microcomputer software market, in contrast, is characterized by the need to appeal to a large number of potential purchasers (a "mass market"), who will not be willing to assume the cost of training to overcome poor user interface design. Thus, user interface design is critical in successful microcomputer product development.

39. Dr. Liddle further asserts in his declaration (at ¶ 37) that the well-known Fred Brooks "rule of thumb" for scheduling software tasks (published in *The Mythical Man-Month*) is inapplicable to "many, if not most, PC [microcomputer] programs", because they are often written by "smaller teams" following different "dynamics" of development. (*Id.* at ¶ 36)

40. I am familiar with the "small team" paradigm for software development, which was characteristic of the entrepreneurial origins of the microcomputer software community. Indeed, in an article I have recently written, I argue that the "small team" approach can effectively be applied to the development of what would traditionally be viewed as large-scale systems applications.

41. It is true that in the "small team" approach, the initial design phase is relatively less important than in the traditional Brooks model, because the more flexible "small team" can adapt more readily to ongoing design changes. That same fluidity may make it more difficult to quantify the degree of effort expended exclusively in the design of a product's functionality and user interface.

42. However, neither of these factors diminishes the overall significance of the user interface design to "small team" software development. Rather, consideration of user interface design issues should, and frequently does, pervade the entire process. Whether the user interface design is initially specified in detail by one team and handed off to other teams for implementation, as under the traditional model, or evolves through the continual attention of a single integrated team, the point remains that user interface design is the most difficult, creative, and critical part of software development.

43. In my own experience in computer software development for a variety of computational platforms, including microcomputers, well over half of the development effort has been spent in the design of the program's functionality and the user interface that expresses that functionality.

44. Neither can the significance of the user interface design be measured by the time spent in "fine-tuning" the details of design towards the end of the process, as Prof. Kieras suggests in his declaration (at ¶ 48). As stated above, successful user interface design should pervade the entire development process. Moreover, even the "fine-tuning" of "detail" can require significant creativity and judgment, if the product is to be of commercial quality. To a significant extent, successful design is the cumulation of just such details.

45. Moreover, I disagree with that portion of the same paragraph in Prof. Kieras' declaration in which he interprets Mr. Kapor's description of the development of 1-2-3 as suggesting that the design of the 1-2-3 menu structure was a "detail" addressed solely at the end of its development. I also believe Dr. Liddle exaggerates in his declaration (at ¶ 41) when he describes the effort as "merely the identification and organization of command words, which was done at the very end of the development cycle."

46. As I interpret Mr. Kapor's description, both the determination of 1-2-3's functionality and the particular expression of that functionality in its user interface and menu structure apparently consumed much, if not most, of Mr. Kapor's (and others') efforts throughout the development process.

Coding, "cloning", and "emulating"

47. It is Borland's witnesses, not Prof. Galler, who confuse the time spent on coding a product with its significance to the final product. For example, the importance of the user interface or the effort involved in its design has almost nothing to do with the obvious fact that some programmed code must generate it, as Dr. Liddle suggests in his declaration (at ¶¶ 42-43).

48. If the program's range of functionality and user interface have been determined in sufficient detail, writing the program code tends to be a routine effort for professional programmers (although the problem-solving skill of the programmers clearly affects the efficiency of the resulting code).

49. If the programming is inadequate to provide an acceptable level of performance—in terms of response time, memory usage, or hardware requirements—the code may negatively affect the user's perception of the product. At the same time, good code cannot overcome a poor user interface. The user interface remains the critical factor in users' perceptions.

50. Moreover, to recognize that programming code can be a time-consuming and difficult task under certain circumstances does not invalidate the point that the most significant *creative* activity and effort in software design is expended in creating the user interface, which is both my opinion and, as I understand it, the gist of Prof. Galler's.

51. In his declaration (at ¶¶ 45-50), Dr. Liddle also quarrels with Prof. Galler's statements regarding the relative difficulty of developing "clone" software, on the grounds that the programming of a "clone" would still need to achieve an acceptable level of performance for the product to be "economically viable" and that Borland's products are not, in his view, "clones" under Prof. Galler's definition.

52. These statements miss the point. Whether a "clone" product would be "economically viable" would depend on a number of factors, including price, the extent of marketing expenditures, the nature of the relevant market(s), and the "clone" developer's cost structure, in addition to the performance of the code for a "clone". For example, a "clone" product with relatively poor performance could be "viable" if its price was a small fraction of the original's, if a market of potential purchasers existed whose preferences would be satisfied by substituting price for performance, and if the "clone" developer's volume of sales could support its cost structure.

53. Moreover, given existing software development tools and practices, it is certainly not conceptually difficult to generate the code for a "clone", or for a program that "emulates" the user interface of an existing product by displaying the same menu structure and accepting the same commands (even

if other aspects of its visual appearance may differ). Development of an "emulation" product would be even less difficult if an adequate computational engine already exists to which the "emulation" commands could be readily "mapped", such as that provided by the Quattro and Quattro Pro "Menu Builder" and "Menu Equivalent Command" facilities. Nothing that Dr. Liddle (or Professors Kieras or Olson) states is to the contrary.

54. The maker of an "emulation" product, like the "clone" maker, would still derive great benefit from the development effort expended by the creators of the original software, since it can avoid much of the effort in functional specification or creation of the "emulation" user interface.

55. The maker of an "emulation" product, like the "clone" maker, also gains the advantage of reducing its market risk, since it already knows that a market exists for software having the characteristics of the original product, and knows that a pool of potential purchasers exists who are familiar with the original product's menu structure and commands.

56. That familiarity would make it more likely that users of the first product would purchase the "emulation" product, even if their ultimate objective is to take advantage of any additional functionality the new product might provide. Users of the first product will perceive lower "transition costs" to themselves in switching to the new product because they will believe that their familiarity with the original product's commands can be applied to the second product, providing both some assurance that they will be able immediately to perform the work they had done with the first, and likely making it easier for them gradually to learn any new commands associated with the second product's additional functionality.

The menus and menu structure of 1-2-3

57. Programs to permit the electronic construction and manipulation of spreadsheets, such as Lotus 1-2-3, have become widely accepted in business, industrial, scientific and technological settings. They are used to enhance productivity

in many white-collar tasks associated with the manipulation and presentation (in both tabular and graphical form) of numerical, financial, and statistical data. Indeed, spreadsheet programs have become invaluable in performing complex calculation, analysis, and modeling, and in forecasting using projection techniques.

58. I am an experienced user of microcomputer spreadsheet software, having used VisiCalc, 1-2-3, and Excel extensively for purposes of modeling, analysis, and graphical presentation. I regularly use spreadsheet software for such tasks, as well as the creation and management of small databases (even including my address/telephone list), and the regular preparation and maintenance of class lists and grade reports. I have taught courses using each of the three programs.

59. I am familiar with 1-2-3, Releases 1A, 2.0 and 2.01. In addition to my use of 1-2-3 as described above, in preparation for this testimony I have reviewed each of these releases of 1-2-3 in operation and reviewed their user's manuals.

60. 1-2-3 is a business productivity software program for microcomputers that integrates spreadsheet, graphing, and database capabilities.

61. In my opinion, 1-2-3 presents the user with a clear and easy approach to operations associated with the construction and manipulation of spreadsheets, databases, and related graphs.

62. In my opinion, 1-2-3's user interface includes numerous discrete elements, including its menus and menu structure, that reflect creative choices and subjective decisions by its developers.

63. In my opinion, the user interface of 1-2-3, including its menus and menu structure, is distinctive among those that could be created to express the idea of a spreadsheet program.

64. 1-2-3's menus and menu structure, in my opinion, convey a substantial amount of information and explanation to the user, giving them an important communicative and

expressive aspect. I believe this to be true even if the "long prompts", or second (lower) line of the 1-2-3 moving cursor menus, are excluded from the analysis.

65. The menu commands and menu structure of 1-2-3, contrary to the statements of Borland's expert witnesses, do not "perform[ ] numerical calculations" (Dr. Liddle, ¶ 51; Prof. Olson, ¶ 28) or "define[ ] the steps or procedures users may follow to perform numerical calculations". (Prof. Kieras, ¶ 35) These statements reflect a fundamental misapprehension concerning the nature and role of the 1-2-3 menu commands.

66. To begin with, the menu commands do not, themselves, provide a means for users to invoke the program's capabilities with respect to numerical calculation, nor do they "define the steps or procedures" a user must employ to perform such calculations. Those calculations are indicated by the user to the program through the inclusion of formulas, "@ functions", and arithmetical operators (such as the "+" or "-" sign) in the cells of the spreadsheet. The menu commands, in contrast, primarily concern the manipulation of data and text in the spreadsheet.

67. For example, Prof. Olson's characterization in her deposition (at Vol. 2, pp. 55-56 & 82) of alphabetical sorting as a form of "numerical calculation" is erroneous and reflects little understanding of either standard programming techniques or the internal workings of a program such as 1-2-3. (Copies of pages referred to throughout this declaration from Prof. Olson's deposition are attached hereto as Exhibit B.)

68. If Borland's witnesses genuinely intend to suggest that the 1-2-3-menu commands constitute some type of complicated method for users to perform numerical calculations such as multiplication, they are completely in error.

69. Moreover, the misstatements reveal that Borland's witnesses are confusing the program's underlying functionality with the particular expression of that functionality in its user interface. The user interface and menu structure of a software program, however, have a distinct existence and purpose.

70. The words in the 1-2-3 menu structure do not actually perform or instruct the computer to perform particular operations. Before the program will begin to perform an operation, the user must take an action—such as typing a series of keystrokes—that will then cause the software to instruct the computer to work.

71. It is inaccurate to equate, for example, the screen display of the word "Print" with the electronic processes whereby the program instructs the computer to cause certain information to be physically printed. The purpose for displaying the word "Print" on the screen is purely communicative in nature. In the context of the 1-2-3 menu structure, the word conveys to the user that one available type of operation that the user can instruct the computer to perform pertains to the program's printing capabilities. The program need not convey any such information on the screen in order to provide those functional capabilities, as would be true, for example, with a command-driven user interface.

72. That the words in the 1-2-3 menus may, in some circumstances, provide a lesser quantity of information than the long prompts or the documentation or the "help" screens does not imply that they are less communicative in nature. The difference is one of degree, not of kind.

73. It is possible to overstate even the difference in degree between the words in the 1-2-3 menus and the long prompts. To take a typical example, it is questionable how much additional information is conveyed by the long prompt "Set global format" beyond that already conveyed by the menu commands "Worksheet/Global/Format". It cannot meaningfully be said that the former is explanatory (and therefore communicative in nature) while the latter is not.

74. Neither is it accurate to suggest, as do Borland's witnesses in their declarations (Prof. Kieras, ¶ 38; Dr. Liddle, ¶ 52; Prof. Olson, ¶ 32), that the words in the 1-2-3 menus "identify the functions that are available in the system" in the sense that they are merely markers or placekeepers for particular commands. Rather, the words in the menus provide a

form of structured dialogue that helps to *inform* the user about the nature of each command.

75. Moreover, each word in the 1-2-3 menus does not correspond to a particular primitive of underlying functionality. Selecting the "Worksheet" menu item, for example, does little other than cause the program to display the next set of choices in the branching sub-menu. The "function" of that menu item is to tell the user that if he selects this, another series of choices will become available that will affect the "Worksheet". That is communication, not calculation.

76. It is this same interactive and dynamic nature of the menus that distinguishes the words in the 1-2-3 menus from the buttons on a VCR or any other set of static controls for a machine or device.

77. The expressive nature of the 1-2-3 menus is not diminished by the fact that sophisticated users often memorize frequently used command sequences, and do not need to refer to the menus to invoke them. Part of the brilliance of the 1-2-3 design is precisely that it allows users to learn the menu commands gradually, and to increase their skills incrementally with use. The words in the menus and the menu structure undeniably assist in that learning process. In addition, even sophisticated users may need to refer to the menus for assistance in unfamiliar parts of the menu tree, to select commands that they do not frequently use.

78. Thus, Borland's witnesses are incorrect when they suggest in their declarations (Dr. Liddle, ¶ 51; Prof. Olson, ¶ 31) that the 1-2-3 menu commands and menu structure are "inseparable" from 1-2-3's functionality, or from a "command system" providing access to that functionality. Any word or symbol could, as a technical matter, be used instead of the words used in the 1-2-3 menus to represent functionally comparable commands, and such words or symbols could be arranged in any number of menu structures that could be "mapped" onto the same underlying functionality.

79. I see no distinction in this regard between the role of words in menus on the one hand, and of icons or other pictorial matter on the other. Users will still associate a specific icon in a program with one of its specific commands, and subtle differences in the expression of icons could inhibit users' ability to learn a new user interface in the same way that the use of similar but different words could. If the words in the 1-2-3 menus are "inseparable" from its functionality, so too would icons be "inseparable".

Alternatives to the 1-2-3 menu structure

80. In my opinion, the menus and menu structure of a successful spreadsheet program could be very different from those of 1-2-3. This would even be true of a spreadsheet program performing all the same functions as 1-2-3.

81. Borland's witnesses do not, as I read their declarations, directly refute this point. It is clear that when they fault Prof. Galler in their declarations for making an assertion similar to mine in the preceding paragraph (Dr. Liddle, ¶ 67; Prof. Olson, ¶ 50), they do so only because they have artificially defined the functionality of 1-2-3 to include the user's ability to execute, modify, and "debug" macros previously written for use with 1-2-3 in another program, *without* having to learn a different menu structure. (*See* ¶¶ 115, 134-139, below.) Indeed, Prof. Olson admitted at her deposition (at Vol. II, pp. 157-158) that Prof. Galler's statement would be true if only the objective of executing 1-2-3 macros in another program were omitted.

82. Borland's witnesses also assert in their declarations (Dr. Liddle, ¶ 54; Prof. Olson, ¶ 35) that the potential number of words or synonyms in the English language that could sensibly be chosen to convey the meaning of each word used in the 1-2-3 menus to represent a menu command, taken individually, is relatively limited. I do not believe that this point, even assuming it to be true, suggests any meaningful limitation on the potential variation in spreadsheet user interfaces or menu structures.

83. To begin with, it assumes an unnecessary constraint of using words to represent menu choices rather than icons or other symbols. It also assumes that the organization of the menu tree is fixed, so that the developer's only option is to find synonyms for the same set of menu commands.

84. Without the latter constraint, it is clear that the potential variation in menu structures is enormous. Assume that a program contained only nine items at the leaves, or nodal points, of its menu tree. The number of possible arrangements of that set into one or more menu levels (including the possible combination of two or more items into a single item) exceeds 1,000,000,000. The number increases markedly with the addition of each item to the set. The number of possible arrangements for a ten-item set would exceed 38,000,000,000. I have not calculated the number of possible arrangements for a set of 469 menu items, but it certainly vastly exceeds the estimated number of elemental particles in the universe which, if my memory serves me, is on the order of 10^{80}.

85. Even if one accepts the arbitrary constraint of allowing for only one such arrangement of menu items, simply allowing for the choice of several potential *synonyms* to represent each such item still produces an enormous potential variation in menu trees. The total would be derived by the simple formula s^i, where i represents the number of menu items and s represents an assumed constant number of potential synonyms. Applying this formula to the 1-2-3 menu arrangement and assuming three possible synonyms for each menu item, the number of possible variations in the 1-2-3 menu tree would be 3^{469}, or 10^{223}. As large as this number is, it is less than the number of possible rearrangements of the items in the 1-2-3 menu tree described in the preceding paragraph. When one takes into account both sources of variation (structure and synonyms) the number grows explosively.

86. Moreover, a successful spreadsheet product could differ from 1-2-3 in its functionality. The potential variation in menu trees that could result from possible differences in functionality is literally unknowable.

87. The validity of the point is confirmed by ample "existence proofs". Numerous examples can be found of programs that provide spreadsheet functionality comparable to 1-2-3 but contain user interfaces, including menus and menu structures, differing from 1-2-3. As I mentioned in my testimony in *Paperback*, these would include Microsoft's Excel, Javelin, Lotus' Symphony, SuperCalc4, MultiPlan, VisiCalc, and others.

88. Excel is similar in idea to 1-2-3—that is, it permits the user to perform the same basic functions incident to constructing and manipulating a spreadsheet. However, its user interface, including the structure, organization, and content of its menus and menu tree, is quite different from that of 1-2-3.

89. Javelin is a product that provides an entirely different user interface and approach to most of the tasks that spreadsheet programs are used to perform, such as the use of mathematical functions and the ability to perform the manipulation of numerical and statistical data.

90. Symphony is an integrated product incorporating five functional components, performing spreadsheet, word processing, communications, database, and graphics tasks. The menus and menu structure of Symphony's spreadsheet component are different from Lotus 1-2-3, yet provide functionality comparable to that provided in 1-2-3.

91. Indeed, Borland's own Quattro and Quattro Pro products are self-contained demonstrations of this point. Each provides a so-called "123-compatible" menu tree or "emulation" mode, and at least one other that is different from 1-2-3. The documentation for each product provides tables (Table 4.3 in the Quattro "Reference Guide" and Table 3.3 in the Quattro Pro "@Functions and Macros" guides) comparing the menu commands in the different menu trees to the so-called "Menu Equivalent Commands", which constitute Borland's own form of macro notation. Viewing the Menu Equivalent Commands as a specification of the functional "primitives" contained in the products, these tables demonstrate precisely how differ-

ent menu arrangements can readily be "mapped" to the same underlying functionality.

Is the 1-2-3 menu structure a "system"?

92. I have already noted that Borland's witnesses err in describing the 1-2-3 menu commands and structure as a "system for performing numerical calculations", because they neither perform calculations nor are they the means by which the user instructs the program to perform calculations. I also believe that their use of the term "system" is inappropriate in this context and, ultimately, beside the point. Labeling the menu items and menu structure as a "system" does not alter their essential nature as a communicative medium.

93. "System" is a term with many usages and varying definitions, both within and among different disciplines. For example, Dr. Liddle describes a system in his declaration (at ¶ 25) as "a set of actions and results defined in particular relationships to one another." Prof. Kieras, in his deposition (at Vol. 1, p. 159), defined a system as "a set of things that have relationships with each other." (Copies of pages from Prof. Kieras' deposition referred to throughout this declaration are attached hereto as Exhibit C.)

94. In my own work as a systems designer and analyst, I have previously defined a system as "an entity composed of related parts directed at a purposeful activity."

95. Thus, in systems analysis, one could describe an organization comprised of various coordinated subunits as a "system" for manufacturing, distributing, and selling goods, or an integrated collection of computer hardware and software as a "computer system" for processing information.

96. A "system" almost always has parts that themselves could be described as "systems" or "subsystems", and is almost always itself a part or "subsystem" of a larger "system"

97. Every "system" is defined essentially arbitrarily for the purpose at hand. What might be called a "system" from one

perspective could also be viewed as a component of a higher-level "system".

98. A system's "boundary" defines the activities considered to be integral parts of the system, and everything not included within the boundary is its "environment". For example, the president of a large organization could legitimately consider the entire organization as a system, whereas a plant manager within that same organization would likely view its subunits other than the plant as part of the plant's environment.

99. It is possible to describe an application computer program, comprising various modules and subroutines, either as a "system" for providing certain computational capabilities, or as a "subsystem" of the "information system" of which it is a part.

100. Indeed, any computer program that is broken down into more than one monolithic part could be viewed as a "system". Software development is an archetypical systems design problem.

101. In my opinion, it is a misuse of the term to describe the words in the 1-2-3 menus and their organization, standing alone, as a "system". To the extent that the definition can be stretched to apply to them, they are a "system" only in the same sense that a textbook or even a poem would be. It adds little to our understanding of menus, however, to describe them as a system.

102. An essential characteristic of a system is that it accepts input upon which it operates in a functional manner to generate output. The 1-2-3 menus, themselves, neither accept input nor generate output. They are, instead, a communicative medium that conveys information to the user to allow the user to provide input to the computational engine and to cause the computational engine to generate output.

103. Moreover, the "function" of the words in the 1-2-3 menus is to communicate. That is an end in itself and is not,

ordinarily, the type of functional activity to which the term "system" is applied.

104. If the term "system" *could* be applied to the communicative 1-2-3 menus and menu structure, by the same reasoning a textbook organized into related chapters would also constitute a "system" for learning a subject matter, or a poem consisting of integrated lines and stanzas could be described as a "system" for enlightenment or entertainment. In fact, I have used the example of a textbook to illustrate how humans deal with complex problem-solving by breaking the problem down into a hierarchy of sub-problems. This is the same sort of creative act required to design an effective menu structure.

105. Again, however, the classical systems characteristics (for example, "coupling" between subsystems—when the output of one subsystem serves as an input to another subsystem—and the use of shared resources) have no operational meaning in the design of menus.

106. In all events, I do not see how an attempt to apply the "system" label to the 1-2-3 menu structure assists the analysis here. I note that the definition of "computer program" in § 101 of the Copyright Act is "a set of statements or instructions to be used in a computer in order to bring about a certain result." This fits the definition of a "system" supplied by Borland's witnesses (and my own). Thus, as a matter of logic (and not of law), it seems to me that the definition of what is or is not a "system" cannot alone help to determine whether 1-2-3's menus and menu structure are legally protected, since any "computer program" as defined by the statute can validly be described as a "system".

107. Moreover, viewing the 1-2-3 menu commands and structure as a "system" implies no more with respect to the ability of other programmers to design spreadsheet programs with distinct menu commands or structures than it does (or did in *Paperback*) to describe them as forming a programming language.

108. If the 1-2-3 menu structure could be described as a "system", so could all menu structures, indeed all user interfaces, including those that use icons or other symbols in their menus. There could then be as many different "systems" in that sense as there could be alternative user interfaces, each defined by the program from which it arises: an Excel "system", a native Quattro Pro "system", etc. Describing menu structures as "systems" has no bearing on the potential variation in user interfaces or menu structures for spreadsheet programs providing comparable functionality.

109. Describing and studying user interfaces as "systems" seem immaterial to me for understanding their inherently communicative nature.

The commercial value of the 1-2-3 menus and menu structure

110. In my opinion, when 1-2-3 was first introduced, its user interface, including its menus and menu structure, contributed substantially to 1-2-3's success. I personally switched from VisiCalc to 1-2-3 at this time because of its enhanced features and its powerful and "friendly" interface, which was a major reason for my willingness to learn a new product.

111. The user interface of 1-2-3, like those of other successful microcomputer software products, has great commercial value.

112. The menus and menu structure of 1-2-3 are, beyond doubt, a critical component of its user interface and themselves have great commercial value. They are critical in forming the product's identity to the user.

113. Whether other words could have been chosen or another arrangement found for the 1-2-3 menus at the time of its development, without significantly affecting the degree of commercial success that the product achieved, is a question that cannot be answered empirically. But this is, in my opinion, an irrelevant question. Once users did learn the 1-2-3 menu commands and came to identify them with Lotus 1-2-3, those particular commands and their arrangement clearly

acquired great significance. Their *current* value, to both Lotus and Borland, lies precisely in the familiarity that millions of 1-2-3 users have acquired with them in the more than eight years subsequent to the product's introduction.

Macros and macro compatibility

114. In the *Paperback* case, the defendants posited a "requirement" of "macro compatibility" that, they argued, compelled them to duplicate the 1-2-3 menu structure. Defendants contended that "macro compatibility" was necessary in order to permit 1-2-3 users to retain the benefits of macros developed or purchased by them for use with 1-2-3 when switching to another spreadsheet product. Defendants also asserted that "macro compatibility" with 1-2-3 was necessary in order to sell a spreadsheet product.

115. Borland's witnesses here present a variation on this theme, apparently contending that it was necessary to duplicate the 1-2-3 menu structure not to execute macros written for use with 1-2-3, but to modify and/or to "debug" (*i.e.*, to fix errors in) them. (Dr. Liddle, ¶¶ 64-67; Prof. Olson, ¶¶ 47-50)

116. As I testified in *Paperback*, in my opinion the entire issue of how much "compatibility" one program should provide with another is, at bottom, a design decision based upon marketing concerns.

117. In the context here of achieving "compatibility" with one or more existing and competing products, the goal of such a decision is to reduce the transition costs to potential customers of switching to another product, in order to be able to persuade more potential customers to buy the new product.

118. There are various transition costs involved in switching from use of one product to another. The nature of these costs may vary from user to user.

119. Some users, who have stored quantities of data using the original program, would incur a transition cost if they had to re-enter on a keyboard those data to use with a new program. A developer may therefore try to make it easier for such users to adopt the new program by eliminating that need.

120. The conventional method for doing so is to provide "file compatibility"—*i.e.*, the ability to read, interpret, and use data stored in the format of files created using another existing program.

121. By providing "file compatibility" with a significant number of existing formats, a developer can sufficiently overcome most user's concerns about transition costs so as to remove such concerns as an inhibiting factor on the user's purchasing decisions.

122. "Macro compatibility" between software programs is less frequently encountered. I understand it to mean the ability of one program to read and interpret "macros", or stored sequences of commands, created using another program.

123. "Macro compatibility" is different from file compatibility. A program can have the ability to read and interpret data, expressions, formulas, and certain attributes written in another program's file formats without having the ability to interpret that other program's macros.

124. The provision of "macro compatibility" is not a technical constraint inherent in the design of a spreadsheet program. Different spreadsheet programs can provide comparable functionality without providing any degree of "macro compatibility" with other programs.

125. Thus, "macro compatibility" itself reflects a design choice made for marketing reasons, presumably to lower the transition costs a user might incur if previously-stored macros would need to be rewritten for use with a new program product.

126. It is not necessary to duplicate the user interface or visible menu structure of another program in order to provide "macro compatibility".

127. The ability to read and interpret macros created using another program can in principle be achieved by a conversion

or translation program, whereby the symbols and syntax of the first program's macros are translated into a form that the second program can understand.

128. Conceptually, all macros could be translated or converted, but as the level of macro complexity increases the design complexity involved in achieving the conversion also grows. Eventually the point is reached where design trade-offs favor leaving some of the responsibility for the conversion to a human analyst.

129. Macro conversion programs can be both effective and practical for translation of a majority of macros, many of which are fairly simple. For example, Microsoft's Excel provides an effective utility for converting Lotus 1-2-3 macros.

130. From my observation and analysis of the Quattro and Quattro Pro "Menu Equivalent Commands"—to which both the "123-compatible" and so-called "native" menu trees in each product can be, and are, mapped—it appears that each of the Borland products already contains the internal programming routines necessary for such a translation or conversion. Indeed, when the user of the "123-compatible" menu trees in each product records a macro in "logical" (*i.e.*, Menu Equivalent Command) mode, as opposed to "keystroke" (*i.e.*, 1-2-3 menu tree) mode, such a translation or conversion process must occur.

131. Moreover, Quattro Pro Versions 2.0 and 3.0 contain a feature in their "native" modes called the "Tools/Macro/Key Reader" command, which permits the user to set the program to execute 1-2-3 macros *without* displaying or invoking the "123-compatible" menu tree. This feature simultaneously demonstrates that it was unnecessary for Borland to display the "123-compatible" menu trees in order to provide macro compatibility with 1-2-3, and confirms that the Quattro Pro product contains the internal routines necessary to achieve translation or conversion.

132. In my opinion, the majority of spreadsheet program users do not write macros extensively, if at all. Such users would be relatively indifferent to "macro compatibility".

133. Even those users who do employ macros are likely to be satisfied by the provision of a conversion program such as Excel's that would enable them to transfer most, but not all, previously-created macros to the new program.

134. The ability of users to modify or debug macros written in one program with a second program, in my view, presents an even more remote and theoretical concern on the spectrum of compatibility.

135. If a program contained an effective conversion or translation facility for macros created using another program, that same facility could also effectively provide for the display of menus, dialogue boxes, or other information necessary to modify or debug the converted macros, and could do so using the unique menu structure of that program. Once the translation has been made, the macro has been effectively converted to the new debugging environment (including any debugging facilities included in the new environment).

136. Certain "interactive" macros are written in a manner that causes the macro to pause in its execution in order to allow the user to make a selection from a menu. After that selection is made, the macro resumes execution. A converted macro could, in that situation, cause the program to display the appropriate menu from the unique menu structure of the second program. The technical difficulty in doing so would not be significantly greater than the difficulty in devising the translation facility itself.

137. Of course, the user who is familiar with the menu structure of the first program would need to learn the unique menu commands of the second in order to make the appropriate selection. That is a cognitive problem for the user, not a technical constraint of programming. Any difficulty the user might encounter in that situation is no different from the difficulty he or she would have in learning the second program's

unique menu structure to begin with. Because a user is unlikely to encounter such an "interactive" macro in the second program before being exposed to that program's menu structure itself, in my opinion this issue is entirely subsumed in the more fundamental issue of whether it is relevant to consider the user's potential difficulty in learning a new menu structure in this context at all.

138. Moreover, in my opinion, the ability to debug or modify converted macros by reference to the menu commands of the first program would have very little significance to the vast majority of spreadsheet users.

139. To the extent that macros written by one user are frequently shared in a corporate environment, as Borland's witnesses assert in their declarations (Dr. Liddle, ¶ 61; Prof. Olson, ¶ 44), that "one user" is typically an experienced in-house programmer or member of the department's technical support staff who would also possess the skills and motivation necessary to address any problem in modification or "debugging" that the translation or conversion facility did not automatically solve.

The "123-compatible" menu trees of Quattro and Quattro Pro

140. I am familiar with the so-called "123-compatible" menu trees or "emulation" modes of Quattro, Version 1.01, and Quattro Pro, Versions 1.0, 2.0 and 3.0, having reviewed each program in operation and having reviewed their documents. I have also reviewed the exhibits attached to the Declaration of Laurie Flesher, which were submitted by Borland in this case.

141. In my opinion, the menu commands and structure of each "123-compatible" menu tree are substantially similar in their communicative nature to the menu commands and structure of Lotus 1-2-3, Releases 2.0 and 2.01.

142. From my review, it appears that every item from the 1-2-3 menus is contained in the Quattro and Quattro Pro "123-compatible" menu trees, and appears in the same menu,

menu level and internal order within each menu. The only exception I can find is that the command "Quit", which (when it is used) typically appears as the last item in a menu, occasionally follows the insertion of new menu items in certain menus in the "123-compatible" menu trees.

143. Every "path" or command sequence through the 1-2-3 menu tree is duplicated in the Quattro and Quattro Pro "123-compatible" menu trees.

144. Each of the menu commands from 1-2-3 is not only represented in the "123-compatible" menu trees by the same word, but also represents comparable functionality in the underlying program.

145. In my opinion, there can be no doubt that the "123-compatible" menu trees are based upon Lotus 1-2-3, or that users would so perceive them.

146. Indeed, the "123-compatible" menu trees are designed in such a way that the points of similarity are explicitly identified to the user. The "new" commands in Quattro and Quattro Pro—that is, those commands which are *not* from 1-2-3—are (with trivial exceptions) clearly marked by a "bullet" or shaded-in square following the word in the menu, as the documentation for the products explains.

147. In addition, for all menus in Quattro Pro other than the main menu, the commands not from 1-2-3 are separated from the 1-2-3 commands by a horizontal line across the menu box (with the occasional exception of a terminal "Quit" command). (The main menu, unlike other menus, is displayed horizontally across the top of the screen as in 1-2-3, and thus no such line could be shown to separate its single new command and terminal "Quit".)

148. The numerical count of menu commands recited in each declaration of Borland's witnesses (Prof. Kieras, ¶¶ 66 & 77; Dr. Liddle, ¶¶ 84 & 95; Prof. Olson, ¶¶ 67 & 78) vastly overstates the differences between the "123-compatible" menu trees and 1-2-3 itself. In Quattro Pro Version 1.0, for example, more than 80% of the "thousands" of "new"

commands are contained in six identifiable "clusters" of the menu tree, below the "Worksheet/Global/Default/Hardware", "Range/Output Style/Fonts", "Graph/Options/Legend/Typeface", "Graph/Options/Titles/Typeface", "Graph/Options/Data-Labels/Typeface", and "Graph/Options/Range Customize" commands.

149. The first cluster, of more than 1700 commands, merely specifies screen and printer information. The next four clusters, of more than 1600 commands, serve only to set typefaces for various types of output display. The last cluster, of more than 600 commands, allows the user to alter the colors and shadings in graphs.

150. Moreover, with one exception these clusters diverge from the 1-2-3 menu tree at the third or fourth level of the "123-compatible" menu tree, with the majority of the "new" commands appearing from five to eight (or more) levels deep in the menu tree. Approximately 90% of the total of "new" menu commands in Quattro Pro Version 1.0 appear at least five levels deep in the "123-compatible" menu tree. Furthermore, a great many of these "new" menu commands are merely repetitious commands used in other "branches" of the menu tree.

151. Thus, the integrity and structure of the 1-2-3 menu tree is, so far as possible, preserved intact in the "123-compatible" menu trees of Quattro and Quattro Pro. The 1-2-3 menu commands are not "interspersed at all levels" of the "123-compatible" menu trees, as Borland's witnesses state in their declarations. (Prof. Kieras, ¶ 66; Dr. Liddle, ¶ 84; Prof. Olson, ¶ 67) Rather, the "new" commands in Quattro and Quattro Pro are new leaves hung on the old 1-2-3 branches and stems, which are in turn clearly distinguished for the user.

152. I do not see how Dr. Liddle or Prof. Olson can reconcile the statements in their declarations (at ¶ 87 and ¶ 70, respectively) that the "new commands" are "not merely extra choices that have been tacked on to the Lotus 1-2-3 menu tree", with their subsequent statements (at ¶ 99 and ¶ 84, respectively) that the new "functionality does not really fit

well within any of the Lotus 1-2-3 top level menu categories" but "had for the most part to be placed somewhere within the existing Lotus 1-2-3 top level menus." A fair reading of the assertions in their declarations concerning the "awkward" nature of the "123-compatible" menu trees (Dr. Liddle, ¶¶ 96-103; Prof. Olson, ¶¶ 81-88) is that they are "awkward" precisely because the new commands had to be, and were, "tacked onto" the 1-2-3 menus.

153. The users' exposure to the "123-compatible" menu trees will be concentrated primarily on the top levels of the menu hierarchy, where the differences with Lotus 1-2-3 are not significant. Because users will, as a matter of probability and likely patterns of usage, encounter the top levels far more frequently than lower levels, the top levels will also form the users' impression of the menu tree.

154. Indeed, the six clusters of the "123-compatible" menu tree of Quattro Pro Version 1.0 in which the vast majority of the "differences" with Lotus 1-2-3 are concentrated provide only limited functionality that many, and in some instances most, users will rarely invoke. For example, printer specification commands are likely to be among the least frequently used in the entire menu tree, since most users change their printers very rarely, if at all. Often these choices are made by technical support personnel at the time the software is installed.

155. In addition, Borland's comparisons exclude the "Install" and "PrintGraph" modules of Lotus 1-2-3 from their analysis. These modules provide functionality comparable to much of that provided by the "new" commands in Quattro and Quattro Pro. They are included in the 1-2-3 program disk, are accessible from the 1-2-3 "Access" menu, and have their own menus.

156. The emphasis that each of Borland's experts place in their declarations on the "View", "Window" and "Display" commands in the "123-compatible" menu tree of Quattro Pro (Prof. Kieras, ¶¶ 70-72; Dr. Liddle ¶¶ 88-90; Prof. Olson, ¶¶ 71-73) is, in my view, extraneous. Neither the "View" nor

"Window" commands alters the contents or display of other menus, and the "Display" command alters the display only of one submenu concerning graph types.

157. I believe that the "difference" in "problem solving approach" each of Borland's witnesses purport to divine from the long prompts in 1-2-3 and Quattro Pro Version 1.0, and which they elaborate upon in their declarations (Prof. Kieras, ¶¶ 57-61; Dr. Liddle, ¶¶ 75-79; Prof. Olson, ¶¶ 58-62), is nonexistent. Contrary to the assertion that Lotus 1-2-3 "usually" or "often" displays in its long prompts the next submenu that will appear if the user invokes the currently highlighted command (Prof. Kieras, ¶¶ 57-58; Prof. Olson, ¶ 59), the Lotus 1-2-3 long prompts do so in fewer than ten instances. Four of those instances occur in the top level or main menu.

158. I also fail to discern, in the vast majority of instances, any significant difference in the contents or message of the long prompts in the respective programs. In my view, the Quattro Pro long prompts for 1-2-3 commands, even where the wording is somewhat different, are mostly paraphrases of the 1-2-3 long prompts. This is not surprising, since the purpose of both sets of long prompts is to describe identical commands representing the same associated functionality.

159. Moreover, differences such as that between a display of the submenu branching from the "Worksheet" menu choice in 1-2-3, and the long prompt "Commands that effect [sic] the entire worksheet" in Quattro Pro, are too slight to sustain the weight of a "difference" in an entire "problem solving approach".

160. Similarly, I cannot see that the on-screen display of prior menu choices in the "cascading" menu boxes of Quattro Pro, which occurs to some extent but not always (at times all but the first letter of a prior choice, and occasionally even that, is obscured by a subsequent menu box), evinces a different "problem solving approach".

161. It may be helpful to users to see information concerning prior menu choices because it enables them to keep

track of their context in the menu tree. (At the same time, since the previous long prompts are not displayed, this point further demonstrates that the menus themselves provide valuable information to users.)

162. However, the more important purpose of the menu tree is to point the user to the appropriate *next* selection. The display of prior choices does not materially assist the user in making that selection.

163. In all events, I cannot tell from where Borland's witnesses gleaned the idea that the contents of the long prompts reflected a significant element in 1-2-3's design. Mr. Kapor's only statement on the issue in his first declaration in the *Paperback* case (at ¶ 101) is that he applied no "mechanical principle" in this regard, but made decisions "by a case-by-case consideration as to which type of long prompt would be appropriate to a particular command."

164. In my opinion, the approach to operations associated with the construction and manipulation of spreadsheets, databases, and related graphs presented to the user by the Lotus 1-2-3 menu commands and structure, and by the "123-compatible" modes of Quattro and Quattro Pro, are essentially identical. The differences—such as they are—between the menu trees do not affect this result.

165. In forming my opinions concerning the "123-compatible" menu trees of Quattro and Quattro Pro, I have taken into account and recognize the differences in the visual manner of their display. These center upon the use of "pop-up" and "cascading" menus in Quattro and Quattro Pro, respectively, the use of color, and the display of the long prompts in a separate location on the screen. Some of these differences may be useful, others not.

166. I have also taken into account the fact that Quattro Pro permits the invocation of menu commands by use of an external "mouse" device, in addition to the keyboard.

167. These differences do not, however, affect either the contents of the menus or their communicative nature. They

will not confuse 1-2-3 users or diminish their ability to apply their familiarity with the 1-2-3 menu commands to the "123-compatible" menu trees.

168. To the contrary, the "new" commands in the "123-compatible" menu trees have been interpolated in a manner that will (and was apparently intended to) minimize the interference with what 1-2-3 users already know. Once 1-2-3 users have learned how to install the "123-compatible" menu trees (or to load the program to cause them to be displayed, by typing "Q123" at the DOS prompt), they need to learn nothing new to set immediately to work and can apply all their acquired familiarity with 1-2-3 to Quattro and Quattro Pro. They can then, also, learn the "new" commands associated with such additional functionality as the Borland products do provide through an incremental process.

169. In my opinion, in providing the "123-compatible" menu trees, Borland obtained a tremendous commercial advantage by lowering the perceived "transition costs" to 1-2-3 users in switching to its products. Borland was thereby able to exploit the commercial success of 1-2-3 for its own marketing ends.

Consequences of menu protection

170. I do not consider it my place as an expert witness to make policy arguments or to render opinions on the law, such as those advanced in their declarations by Prof. Kieras (at ¶¶ 78-82) or Dr. Liddle (at ¶¶ 108-113). I was not asked to do so in *Paperback*, and did not.

171. If the Court should entertain opinion testimony in this case, however, concerning the likely consequences to software development or the software industry of providing protection to menu structures such as that in Lotus 1-2-3, or to user interfaces generally, my views are as follows.

172. The "efficiency" arguments in opposition to such protection asserted in the declarations of Prof. Kieras (at ¶¶ 81-82) and Dr. Liddle (at ¶ 111) extend, in this case, only to the

short-run gains to Borland or to users who switch from Lotus 1-2-3 to Borland products. There might well be short-run economies to users in lowering "transition costs", and to Borland (or others) in avoiding much of the cost in developing a truly original user interface, if Borland (or others) were allowed to duplicate the 1-2-3 menu structure. Indeed, that conclusion follows from several points in my previous analysis, and was apparently the basis for Borland's copying of the 1-2-3 user interface.

173. However, a more thorough and long-run efficiency analysis would have to consider the systemic loss from the perpetuation of *inefficiencies* resulting from "design flaws" in the 1-2-3 menu structure—possibly including those Prof. Olson claims to discern in her declaration (at ¶ 80)—and also the loss from the slower adoption of superior and more efficient menu structures—such as, according to Prof. Olson in her declaration (at ¶ 89), the "native" menu tree of Quattro Pro. Prof. Olson suggests this very point in the article I previously cited (at pp. 258-259).

174. Also, if it is true, as Dr. Liddle in his declaration (at ¶ 96) and Prof. Olson in hers (at ¶ 81) contend, that the "123-compatible" menu trees are "awkward, difficult and inefficient" with the inclusion of the additional functionality provided by Quattro Pro, as the capabilities of software continue to expand the implication is that little benefit can be expected in the future from similar attempts to superimpose or "shoehorn" those capabilities onto older structures designed for different products.

175. More important, though, is the long-run effect on incentives to software developers to invest in research and development. If menu structures and user interfaces could be freely copied or emulated, the benefits to their creators would be very short-lived. Significant expenditures in product development and user interface design would become much less rewarding. Continuing advances in programming tools and techniques are likely to decrease the benefits from such investment still further. Prof. Kieras, who proudly claims in

his declaration (at ¶ 48) to have "spent a great deal of [his] professional energies" trying to persuade software developers to devote more effort to user interface design, would find it even more difficult to persuade developers of the utility of his studies and theories.

176. The commercial advantage would then lie with those with the most money to spend on marketing, or to those with the most efficient code production. This would favor the large company over the prototypical start-up, and highly-efficient programming houses such as Japanese computer companies over domestic developers.

177. In my opinion, the user interface of software is what provides the critical value in the contemporary software marketplace. It is also one aspect of software development in which United States developers appear to hold a competitive edge. If it is public policy to provide meaningful protection for innovations in software products under United States law, it is imperative that user interfaces should be protected, including their menu structures.

Executed under the penalty of perjury at New York, New York on November 1, 1991.

/s/ JAMES C. EMERY
James C. Emery

CERTIFICATE OF SERVICE

I hereby certify that a true copy of the foregoing was served on counsel of record for the defendant herein by telecopier and overnight air courier on November 1, 1991.

/s/ [ILLEGIBLE]
Illegible

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 87-0076-K
Civil Action No. 87-0074-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

PAPERBACK SOFTWARE INTERNATIONAL
and STEPHENSON SOFTWARE, LTD.,

Defendants.

And Consolidated *Mosaic* Action.

DECLARATION OF BERNARD A. GALLER

I, Bernard A. Galler, declare:

1. I am Professor of Electrical Engineering and Computer Science and Associate Director of the Computing Center at the University of Michigan. A true copy of my resume and bibliography of my publications is annexed hereto as Exhibit A. I make this declaration as my testimony in the Phase I trial to be held in the two above-referenced actions.

2. I consider myself professionally knowledgeable in the subjects of computer software design and development.

3. For the past 34 years, I have regularly taught both undergraduate and postgraduate courses at the University of Michigan, including Advanced System Programming, the most advanced computer programming course offered in the University.

4. Under my direction and supervision, students in that course have undertaken and completed various practical projects in software design and development. These projects have included the design and development of (i) a program for the student course registration system at the University of Michigan; (ii) prototype software for a console for the Ann Arbor, Michigan, dial-a-ride bus system, in which bus schedules are coordinated by computer; (iii) a computer language to express degree requirements for the University of Michigan; and (iv) a compiler program for the purpose of converting programs written in the computer language BASIC into a form that can be executed by a mainframe computer.

5. I have personally designed and written hundreds of computer software programs for mainframe and personal computers, including compilers, operating systems and various applications.

6. In addition, I have published two books and a number of articles in the areas of computer programming and mathematics, as detailed in Exhibit A.

Microcomputers and their characteristics

7. On the most fundamental level, computers are simply machines with the ability to perform three basic functions, all electronically. They can (i) perform arithmetic calculations; (ii) perform logical operations, such as comparing values to determine whether one is larger or smaller than another and decide what to do next based on the result; and (iii) store and/or display the results, either internally or externally.

8. Although computers have only these few capabilities, they are useful tools for solving problems for the following

reasons. First, by combining "bits" (binary digits, the basic units of information manipulated by a computer), it is possible to represent any information in a form which a computer can process. Second, computers can perform millions of operations in a single second. Thus, they can be used to solve problems too complex to solve manually, as well as to perform tasks that would be repetitious and boring if performed by human labor.

9. Therefore, computers commonly are used to accomplish a wide variety of tasks in business, industry, education, science and technology, frequently involving office functions such as word processing and the generation of financial reports; the control of machinery; the solution of complex computational problems; and the recording and storage of large amounts of information.

10. "Microcomputers", or "personal computers" as they often are called, like their larger counterparts, mainframes and minicomputers, typically have several main parts: first, one or more input devices, to enable the user to communicate data and instructions to the computer; second, a central processing unit, which contains the electrical circuits that control the computer and actually perform the arithmetic and logical functions involved in processing data; third, a main memory, sometimes called "random access memory", to store the data being processed and the instructions employed to process the data; and fourth, one or more output devices, to permit results to be communicated back to the user.

11. Input and output devices sometimes are referred to as "peripherals". Examples of input devices are keyboards, "mouse" devices (which permit the user, without using the keyboard, to move the cursor on the computer screen and to convey instructions to the computer), modems (which receive information telephonically) and optical scanners (which read pages of text). Examples of output devices are video monitors and displays, text printers and graphics plotters.

12. Also among the peripherals are storage devices of varying capacities which employ magnetic media, such as internal and external hard disks, floppy disks and magnetic tapes. These are used to store computer programs and data which the computer is not using at a given time.

Computer software and the user interface

13. To perform any task, a computer must be given instructions. The instructions that a computer can execute consist of code, which the user normally does not see when operating a software program.

14. The code causes the computer to generate the "user interface" according to the software developer's conception and design. In microcomputer software, the user interface typically is displayed on the screen of a video monitor.

15. The "user interface" comprises the way in which the program interacts with the user. Typically, the user interface provides information necessary for the user to communicate with the computer and to instruct the computer to perform particular tasks, within the range of functionality that the software is designed to provide.

16. The user interface may also express to the user a model or approach to the general subject the software addresses, which will help the user to understand the subject and guide the user in solving a specific problem at hand.

17. Computer software traditionally has been divided into "operating systems" and "applications software".

18. "Operating systems" or "systems software" are programs that instruct the computer how to get data and other instructions into, through and out of the computer. Examples of operating systems for microcomputers are DOS, XENIX and OS/2.

19. "Applications software" consists of programs which permit the user to perform specific tasks or "applications".

Commercially available applications software packages for personal computers can be used to perform a wide variety of functions, including word processing (electronic assembly, storage, and editing of texts); database management (storage and manipulation of data according to different specified categories); spreadsheet calculation (calculation and manipulation of data involved in financial analysis, modeling and accounting); graphics (pictorial display and creation of graphs and charts); printing and desktop publishing (arrangement, display and printing of text and graphics in different typefaces and formats); accounting (record-keeping for businesses); communications and electronic mail (transfer of information between different computers); software utilities (organization of magnetic disk storage media and diagnosis and repair of commonly-encountered disk-related problems); education and games.

20. These general categories frequently can overlap, as in integrated software which performs various common business functions, often including word processing, spreadsheet calculation, business graphics, database management and communications. In an integrated product, the same data files are transferable among the various components without requiring the user to reinput the data.

The software development process

21. Software development is a creative process, in which imaginativeness and inventiveness are at a premium.

22. From the outset of the development process, developers must make numerous crucial choices frequently involving subjective judgments.

23. The design and development of original applications software can be difficult, expensive, time-consuming and error-prone.

24. The software development process consists generally of requirements analysis and functional specification; systems

design; coding; debugging and documenting; testing and certifying; and preparing final documentation.

25. In practice, these activities are not necessarily pursued in chronologically distinct or sequential phases. It is common for several activities to occur simultaneously, and for the entire process to be repeated a number of times.

26. What the user sees when the completed software product is run on a computer is the "tip of the iceberg" and does not reveal the magnitude of the preceding development process.

27. Successful software development ordinarily involves various approaches that are tried and discarded when they do not work out as anticipated or hoped. The user of the finished software cannot tell how much time and effort was spent in the pursuit of these "blind alleys" during the development process.

Requirements analysis and functional specification

28. Following proper development procedure, the applications software developer first should analyze exactly what the software is to do and the tasks it must be able to perform. In other words, the developer should select the range of functionality which the application will provide. Once this is determined, a developer typically will write a functional specification for the program describing the intended functionality.

29. As part of the functional specification, the developer should begin to define the software's "externals", that is, the manner in which the program will provide choices, assistance and ultimately output to the user, the general nature and appearance of the program's communications to the user, and the manner in which the program is to receive instructions and data input from the user.

30. As part of defining the software's externals, the developer should create a "mental model", or conception of the program's behavior, which the user is intended to form in operating the program based upon the user's perception of the user interface and externals. This model, along with the communicative content of the program's externals, will later find expression in the detailed creation of the program's user interface.

Systems design

31. The developer should determine how the program is to be organized, comporting with the "mental model" of the application, and should decide which steps in the application typically should precede others. This often is done by constructing a flow chart.

32. The developer should also decide which parts of the program have to interact with which other parts, how much the parts should know about each other and which errors the program should be able to recognize and resolve.

33. The developer should also establish the program's technical parameters, including the optimum size of the program's code, how much of the computer's storage and memory should be allocated to store data at different times, and how fast the program should be able to run.

34. For programs which are intended to permit the performance of many interrelated tasks, the management of large amounts of data or the control of several devices, the problems of design are increased greatly. As a program grows in the scope of its intended functionality, technical problems concerning the interaction of its different parts become more difficult. As more people become involved in the development process, problems of coordination and management also tend to grow.

Coding

35. The developers should then write the detailed source code, *i.e.*, the instructions which, when compiled into object code, can be executed by the computer.

36. Typically, the more detailed the functional specification and design documentation, the less significant the task of coding becomes.

37. Generally, coding should not consume more than one-fifth or one-sixth of the total time spent in implementing a new program.

38. There is a growing trend in software development toward automated coding, as exemplified by "compiler compilers" and "report generators", that is, programs which themselves will generate other programs.

39. Based on my experience with the software development process, it is my opinion that implementation of a good design by coding is the "tail of the dog". That is, it is less difficult and requires less creativity and imagination than developing the design in the first place.

Testing, debugging and documentation

40. The developers must also test and debug (that is, remove errors from) the program, and write its final documentation, including both technical notes indicating how the various parts of the program operate and any manual or instructional materials for intended users.

The importance of design in software development

41. For several years, I have successfully used the following rule of thumb for scheduling time and allocating effort in the development of new software: twice the amount of time and effort is typically required for planning, requirements analysis, development of a functional specification and

detailed systems design, including definition of the user interface, than is required for coding the program.

42. This allocation accords generally with the widely accepted model of the "Software Life Cycle" developed and verified empirically by B.W. Boehm, which separates the development of computer systems into six distinct phases: (i) requirements analysis; (ii) system design; (iii) detailed design; (iv) coding; (v) integration; and (vi) operation and maintenance.

43. This model provides a general principle for new software development that is not specific to any particular type of system and applies equally well to the development of microcomputer applications software as to large-scale systems software.

44. Both the allocation which I generally employ and the model of the Software Life Cycle show that specification and design, including the definition of the user interface, is typically the largest component in new software development in terms of the number of man-hours of labor required.

45. Moreover, it is typical in new software development for the more talented and creative members of the development team to be involved through the completion of specification and design, and for those persons then to pass on the responsibility for coding to the less experienced or skilled.

46. Thus, specification and design in the development of new software is typically far more significant than coding, both in the quantity and quality of resources devoted to it.

47. Errors made during specification and design are, among errors made in the development process, the most costly to repair.

48. However, no matter how successfully the software developer addresses the complexities of the development process, and no matter how well-constructed and technically proficient the program's internal workings may be, the micro-

computer applications developer cannot forget that the end result is typically intended for an audience of eventual users who will likely be less concerned than the developer about such matters.

49. The users' perception of the reliability, speed and efficiency of the program undoubtedly influences their view as to the value of the software. However, the users may not even begin to reach these considerations if they do not first perceive the software as providing effective solutions to their particular problems. Users typically form such a perception based upon whether the software appears to them to be well-designed and reasonably easy to learn and to use.

50. Therefore, in my courses, I teach that the central unifying activity of software development is the determination of the actual needs of the intended users. The goal of satisfying those needs with clear, practical and effective solutions should guide the developer's decisions throughout the creation of the software's functional specification and detailed design, including the definition of the user interface.

The importance of the user interface and its creation

51. Within software specification and design, expression of the user interface is probably the most critical task the application developer faces and that in which the most significant creative activity and effort is expended.

52. The user interface should create the "mental model" in the user's mind that organizes the user's way of thinking about the application or problem addressed by the software.

53. The user interface also conveys the program's direct communications to the user concerning operation of the software, including information communicated specifically to assist and instruct the user—in short, those elements which are most intrinsically expressive and most likely to determine how "friendly" or valuable the user will perceive the software to be.

54. Accordingly, clarity, simplicity and precision are at a premium in creating a successful user interface.

55. Creation of the user interface requires the software developer to make subjective choices, decisions and judgments.

56. Initially, the developer must choose the extent to which the user or the software will determine and control the progress of the program.

57. Various possibilities exist. They can be visualized on a continuum ranging from a user interface in which the program's progress is controlled entirely by the user at one end, to a user interface in which the program's progress is controlled entirely by the program at the other.

58. The greater the amount of control that can be exercised by the user, the more likely the software will be to meet the user's individual needs. Conversely, the greater the control exercised by the program, the less flexible the software will be from the user's perspective.

59. Closest to control by the user are "command-driven" user interfaces, in which the progress of the program is determined by commands chosen by the user and communicated to the program in a vocabulary and syntax which the program will recognize.

60. A "command" in this context is a symbolic representation of a particular instruction or set of instructions to the computer, which the program is designed to interpret. It is not, typically, an instruction to the computer itself.

61. If the developer chooses to employ a command-driven user interface, he or she still must choose the range of commands which the program will recognize. For example, will the program recognize a wide range of commands in an extensive vocabulary and a flexible syntax, or a limited range of commands in a limited vocabulary according to a rigidly-defined syntax, or something in between?

62. Providing a greater degree of control by the program are "menu-driven" user interfaces, in which the progress of the program is determined by the user's selection from among predetermined arrays of choices, *i.e.*, "menus", presented to the user by the computer.

63. Even closer to control by the program are user interfaces in a question and answer format, in which the progress of the program proceeds according to the user's selection of answers to multiple-choice questions presented to the user. Such a format is better suited to applications intended to provide only a few possible variations than to applications intended to permit the user to perform complex tasks requiring a higher degree of flexibility.

64. It is even possible to design a user interface in which the program itself guides the program's progress from step to step, with the user making choices no more significant than striking a key to send the computer to the next step of a predetermined sequence of tasks (for example, a tutorial program).

65. If the developer determines that the progress of the program should be controlled to a large extent by the user, the critical decision at this point is likely to be whether or not to utilize "menus".

66. One disadvantage of a decision to employ a command-driven user interface instead of a menu-driven one is that the user must become familiar with and frequently memorize the vocabulary and syntax of the possible commands which the program will recognize. With a menu-driven user interface, because the computer will display to the user the options available at any given point in the software's operation, memorization of commands or of their permissible sequences is unnecessary.

67. Another advantage to the use of menus is that they can be organized in a structure that will convey to the user the approach or "mental model" to the application the developer has conceived.

68. If the choice is to provide menus, the developer must choose how the commands contained in those menus will be represented on the screen. There is a wide range of possible choices, including words, abbreviations, letters, icons (pictures) and other symbols.

69. The developer must also choose the manner in which the user will communicate menu selections to the program. Various possibilities exist, including: (i) use of the computer's cursor movement keys to identify an icon or word, and selection by use of the "Enter" key; (ii) "pointing" to the textual command or icon with a mouse device and "clicking" the mouse to make a selection; and (iii) typing one or more keys on the attached computer keyboard.

70. If the developer decides that the user will select commands from menus by typing a key or keys on the keyboard, the developer has several possibilities for implementing that choice, including: (i) one or more letters corresponding to the desired command, which could be either the first letter of the word or some other designated letter or letters, indicated to the user by a method such as capitalizing, underlining or highlighting; (ii) a "function key" from among the ten or twelve function keys (F1 through F10 or F12) commonly found on personal computer keyboards; or (iii) some combination of these and other means.

71. The developer must also determine the organization of the menus. Again, there are numerous possible variations. The developer should choose that organization which not only best conveys to the user the developer's approach to the application as a whole, but also assists the user in selecting the commands necessary to accomplish specific tasks.

72. For example, if the number of command options to be made available to the user are few, and the task to be addressed by the application is straightforward, the developer may choose to provide the user with the ability to view all basic command options at once, in a so-called "single-tier" menu.

73. Alternatively, there may be so many options, or the tasks to be performed may be sufficiently complex and varied, that the developer may decide that the options should be presented to the user in a "hierarchical" structure.

74. A "hierarchical" menu structure breaks down and organizes the universe of possible command options into groups of related choices in sequence. Each group of related choices is typically communicated to the user by display of a discrete "menu screen".

75. In a hierarchical menu structure, many or most command options are dependent upon others, *i.e.*, they become available only upon the user's selection of particular commands at previous stages in the sequence. A hierarchical menu structure can be visualized as a "command tree" or "menu tree", with dependent commands "branching" from commands at prior levels.

76. In organizing a hierarchical menu structure, the developer must decide the number of command choices to be presented to the user at each menu level, as well as the number (or "depth") of menu levels. Again, there are various possibilities. For example, the menu tree could be "broad and shallow", with many choices presented at once but with only a few levels. Alternatively, the menu tree could be "narrow and deep", with few choices presented at once but utilizing many levels, or something in between.

77. In determining menu levels, the developer should consider the particular sequences of command options that the menus will provide, thinking through all possible combinations and permutations. The developer must, at a minimum, choose the commands from which other groups of choices will branch, and select and arrange those which will branch from them.

78. There is no objectively "right" or "wrong" answer when organizing the menus for a particular application. The decisions are largely subjective.

79. The menu organization should reflect the developer's particular approach to fulfilling the software's intended purpose and functionality, and—as important—the developer's conception of how the likely users of the software will perceive the final result. The developer should seek to provide to the user in the menu structure an intelligent and hopefully intuitive guide to the program's capabilities.

80. Thus, choices concerning menu organization may be influenced by considerations such as the degree of sophistication of the intended user audience, as well as the anticipated frequency with which the users are expected to use the software. For example, users who have become familiar with the possible combinations might prefer a relatively few menu levels to expedite the decision process. Others might prefer to view just a few options at a time in each menu level, particularly when learning the application.

81. The developer could even decide to permit the user to choose from various possible organizations intended for different degrees of user sophistication, or to select an organization that the user could later change as desired.

82. The developer must also decide the manner in which the menus are to be presented to the user, a decision that will influence the user's perception of the software's ease of use. Once again, various possibilities exist, including the display of menus containing words on the computer screen; the display of commands on a template (typically, a small plastic card with printed words) to be placed on or next to the keyboard; and the display of symbols, letters or abbreviations on the screen, with explanatory references contained in the software's user's manual or other printed documentation.

83. Whether the developer decides to present command menus in a single-tier or hierarchical structure, the developer must decide, for each menu, how the menu should be internally arranged. Alphabetical and various substantively-ordered arrangements are possible. The choice of a particular

arrangement for each menu will depend on the developer's view of the relationship, if any, among the choices in each group and the most effective way to convey that relationship to the user.

84. The developer should also select the content of each individual menu choice, including the specific word, icon or other nomenclature to be used to represent that choice, which in the developer's view most clearly expresses and communicates to the user the meaning of the particular command or functionality the choice provides.

85. In sum, creating a user interface requires the developer to apply discerning judgment in choosing from among a wide variety of options, and to exercise significant creativity in expressing the software's model or approach to the particular application in a manner that will appear logical and easy to use.

86. A successful user interface will seem to the user to be naturally suited to the performance of the tasks for which the software is employed, rather than strange or intrusive.

87. The simplicity and ease of use of a successful user interface can sometimes make that interface seem "obvious" in retrospect, or ideally suited and even necessary to the functions the software performs. In reality, however, such a reaction more often reflects the skill, creativity and effort of the developer.

88. Indeed, the possible variations to user interfaces are limited only by the resourcefulness of developers and, in the case of commands which take the form of words, by the constraints of language.

89. Different software programs can provide the user with the same functionality, using the same computer, video screen and operating system, and yet employ user interfaces that the user will perceive as entirely distinct.

90. For example, a commercially-available software product called HyperCard, designed for use with the Apple Macintosh personal computer, permits the creation of "do-it-yourself" user interfaces. Using HyperCard, a user can create many different user interfaces, employing pictures, text, menus, consecutive screens, selection techniques, and other options, for applications designed by the user, including different user interfaces for applications providing exactly the same functionality.

"Cloning"

91. As used in the context of computer software, a "clone" is a program that imitates another program, mimics its behavior and appears identical to it (or virtually so) in external respects, including the user interface.

92. A developer who produces a clone of an existing software product avoids many critical steps in the development process and many of the attendant risks and expenses.

93. For example, by cloning an existing product a developer can avoid almost all aspects of functional specification, much of the detailed design process, and almost all issues involved in creation of the user interface. In my opinion, the "developers" of a clone can bypass most of the creative and more difficult effort typically required in the development process.

94. The developer of a clone also has an easier task in preparing the final documentation for the product. Typically, documentation must undergo frequent changes and numerous iterations to keep pace with ongoing major and minor refinements to the program; the clone developer avoids this, since he knows exactly how the product will look and behave when finished.

95. Even the difficulties of coding and testing a clone can be reduced, because the programmers already know precisely what the clone program should be able to do, and can refer to

a detailed working model (furnished by the existing product) in case of doubt.

96. A reasonably skilled professional programmer can readily take an existing applications program with adequate documentation, run the program and observe its behavior and, using that as a model, clone the existing program.

97. Even good college students given a sophisticated application program such as Lotus 1-2-3, its documentation and an opportunity to observe its operation, could develop a program that mimics the appearance and most aspects of the behavior of the original.

98. Depending on the degree of skill and effort applied, the finished "clone" product might be more crude than the original in its code structure and of inferior quality in performance, that is, less efficient in its use of memory, slower in speed of operation and plagued by "bugs" (*i.e.*, errors).

99. Nonetheless, if one disregarded constraints of performance and simply set out to duplicate the external aspects of an existing software product (given adequate documentation and opportunity to observe its operation), cloning, as a programming task, is relatively easy to accomplish. Even if the clone developer seeks to make a product equal in performance to the existing product, the task is significantly easier than when creating new software.

Electronic spreadsheet software

100. I have examined numerous different electronic spreadsheet programs that are commercially available for use with microcomputers.

101. The basic purpose of spreadsheet software is to permit the user to collect and arrange large quantities of numerical, statistical or financial data, as well as to perform complex calculations with that data and to manipulate it to view and analyze scenarios based on different assumptions.

102. Spreadsheet software allows the user to represent and create electronically a spreadsheet that otherwise would have to be prepared manually, with handwritten entries and arithmetic or with a calculator.

103. The spreadsheet program causes the computer, according to the user's instructions, to perform all calculation and recalculation automatically, including the application of numerous mathematical, logical, financial and statistical formulas, expressions and functions chosen by the user.

Lotus 1-2-3

104. I am familiar with the operation of Lotus 1-2-3 ("1-2-3"), Releases 1A and 2.0. 1-2-3 is an integrated microcomputer software product providing spreadsheet, graphing and information management capabilities.

105. As will be demonstrated at trial, 1-2-3 employs a hierarchical menu structure to display, in full words of text combined in relatively short menus, the options available for the user to invoke in order to instruct the program to perform particular tasks or operations.

106. 1-2-3's menu structure, appearing to the user in sequences of variable menu screens, presents the user with a particular approach to, or model for, the solution of spreadsheet-related tasks and problems.

107. 1-2-3 employs a method of menu display commonly called the "two-line moving cursor menu". Each menu screen contains two lines. By use of the left and right arrow keys, the user can cause the cursor to move, highlighting different words in the upper, "menu" line.

108. Each time the cursor is moved to highlight a different word in the menu line, the program causes different information to be displayed in the lower line of the menu screen.

109. The 1-2-3 menu lines contain series of words, such as this from the "main", or first, menu: "Worksheet Range Copy

Move File Print Graph Data Quit". When a menu screen is first displayed, the word furthest to the left in the menu line is highlighted, *i.e.*, "Worksheet".

110. Each word in a menu line begins with a different letter and represents a different option then available for the user to invoke.

111. To select an option from a menu line in 1-2-3, the user can either (i) strike the key corresponding to the initial letter of the desired word, or (ii) highlight the desired word by use of the right and left arrow keys and then strike the "Enter" key.

112. The words in the menu lines correspond either to commands to instruct the program to take some action, or to options (such as "Global" or "Range") that primarily serve to introduce subsequent sets of logically-related commands pertaining to particular areas of the program's operation. For example, "Global" introduces a set of commands that can be used to set parameters for the spreadsheet (or "worksheet" in 1-2-3's terminology) "globally", or as a whole.

113. The words contained in 1-2-3's menus are not the only words that could be chosen to correspond to their respective commands or options. From my experience, in most instances there are many others that could sensibly be employed in their place. Moreover, the choice of word to represent a command or option in a menu line has no direct effect on the program's underlying functionality.

114. The lower lines of 1-2-3's menu screens display one of two kinds of information: (i) either a "submenu", *i.e.*, the next menu level or set of choices that will become available if the user selects the highlighted word from the current menu line; or (ii) an explanatory comment about the highlighted word in the menu line.

115. For an example of a submenu in the lower line of a menu screen, when the word "Worksheet" is highlighted in

1-2-3's main menu, the lower line displays the words "Global Insert Delete Column Erase Titles Window Status Page". These words will appear in the menu line of the next menu screen the program will display if the user, in fact, selects the "Worksheet" choice.

116. For an example of an explanatory comment in the second line, when the word "Copy" is highlighted in the main menu, the message "Copy a cell or range of cells" appears in the lower line.

117. In no instance does the lower line display commands or options currently available for the user to select. In my opinion, the purpose of the lower lines is solely to provide the user with information concerning the highlighted options in the menu lines, and to apprise the user of the consequences of their selection.

118. 1-2-3's two-line moving cursor menus and hierarchical menu structure effectively give 1-2-3's user interface an informative dimension like that of a thumbnail user's manual, especially for new or infrequent users. Adding to this effect, 1-2-3 also incorporates an on-line, context-sensitive "Help" facility from which the user may obtain basic instruction at almost any stage in the program's operation.

119. In my opinion, the 1-2-3 menus and menu structure clearly reflect their developers' approach to the solution of spreadsheet problems. They appear simple, natural and straightforward. Far from suggesting that they were therefore easy to create, this implies to me that the developers of 1-2-3 expended considerable effort, planning and creative thought.

120. However, 1-2-3's user interface is not the only way, and not necessarily even the best way, to articulate a model or approach for a spreadsheet program, even assuming a design decision to employ a menu-driven user interface.

1-2-3's macro commands and function keys

121. 1-2-3 macros, at their most basic, are abbreviations for user-defined series of keystrokes. Each macro represents a string of commands arranged in a particular order or syntax.

122. After recording and storing a macro, the user can later instruct the computer to perform the entire series of keystrokes by retrieving and "executing" the macro, *i.e.*, striking the key or keys which invoke the macro.

123. 1-2-3 also contains a set of commands listed in the 1-2-3 user manual and referred to as "/X" commands, which are used in the creation and execution of macros.

124. The set of commands to which the "/X" commands correspond is one set of possible choices from among many that could be used to perform comparable functions.

125. 1-2-3 also permits the user to communicate certain frequently-used commands to the program by use of the computer keyboard's special "function keys". For example, F1 corresponds to "Help", F2 to "Edit", and F5 to "Goto".

126. 1-2-3's selection of the functionality to assign to the function keys as a whole (as opposed, for example, to the menu commands); its selection of which commands to assign to different keys: and its selection of words or names to represent those commands, are not dictated by the computer hardware or by any other constraint. A spreadsheet need not use the function keys at all, much less assign to them any particular functionality, command or name.

Alternatives to 1-2-3's user interface for a spreadsheet

127. If a software developer were to undertake to create a user interface for a spreadsheet program that would be used for the same general purpose as Lotus 1-2-3 (*i.e.*, the construction and manipulation of spreadsheets), the ways in which such an interface could be expressed are essentially unlimited.

128. At the outset, there is no inherent reason why a spreadsheet program need include all of, or the same, functionality that Lotus 1-2-3 provides. While certain operations (such as, for example, the basic arithmetic operations) are probably integral to spreadsheet usage, a spreadsheet could provide the user with the ability to perform fewer, more or simply different operations than 1-2-3.

129. Moreover, there is no technical constraint compelling even a spreadsheet program providing the same functionality as 1-2-3 to do so by using the same set of commands for communications between the computer and the user.

130. And even if a developer intended to create a program that provided precisely the same functionality as 1-2-3, there is no inherent or technical reason requiring its user interface to resemble, much less approximate or duplicate, the appearance and expressive characteristics of 1-2-3.

131. To the contrary, it is currently a common and even preferred technique in software development to implement that portion of the source code which generates the user interface (the "presentation manager") separately from the underlying "computation platform", or core of the program which performs the basic operations and data processing. The purpose is specifically to permit the developer to replace or modify one with minimal effect upon the other, so that, for example, changes could be made to the program's internal workings without causing an apparent change in what the user perceives of the program.

132. At the same time, this technique makes it readily possible to erect an entirely different user interface, or more than one user interface, on the same underlying platform. The developer needs only to "map" commands from the new user interface, whatever they may be called and whatever the manner of their presentation to the user, to the appropriate functionality of the computation platform.

133. Thus, the possibilities for creating alternative user interfaces even for programs that provide the same functionality are virtually endless.

134. A spreadsheet program need not employ menus at all. It is possible to design a user interface for a spreadsheet program that does not employ menus.

135. Even if menus are employed, moreover, there is no inherent or technical reason for a developer to duplicate or imitate the 1-2-3 menu structure. It is possible, for example, to design a menu structure for a spreadsheet program that would present command options in only one or two menu levels, rather than the four or five 1-2-3 frequently presents.

136. Even with a menu structure of comparable depth, it is possible to design a structure very different from 1-2-3's. There is no technical reason for any given menu level to contain words or symbols representing the same command choices as those of 1-2-3, or for the dependent command choices in lower-level menus to branch from precisely the same commands or options as they do in 1-2-3's menus.

137. It is possible to design a menu structure in which different sets of commands would branch from different initial menu choices. For example, there is no requirement that a menu option which introduces the commands permitting the user to control worksheet-wide settings (comparable to the "Global" command in 1-2-3) be located in a second-level menu, rather than the main menu or any other. (Indeed, there is no requirement that there be such a command at all.)

138. Moreover, neither the particular order nor choice of words within 1-2-3's menus is dictated by any technical requirement of the operations performed when those commands are invoked. It is possible to design a user interface with menus of pictures or "icons" instead of words; letters; other words to represent each command; or abbreviations of words, such as "Univ" (for universal), "Tab" (for tabulate), "Info" (for information), "Tr" (for transpose), and so forth.

These could be arranged in any order without affecting functionality.

139. It is also possible to create a user interface with a way of invoking commands other than typing the first letter of the word corresponding to the command or by highlighting the preferred selection with cursor movement keys and striking the "Enter" key. Therefore, the potential universe of options is not confined to words with particular first letters, a fact which provides a developer with substantially greater flexibility.

140. There is no technical requirement that a spreadsheet program incorporate a means of communicating explanatory information about each menu choice to the user. But if a developer did decide to incorporate a means of communicating such information, it is certainly possible to design and incorporate a means of communication that would be different from the informational lower line of 1-2-3's menus.

141. For example, a program could display information about highlighted menu choices in a special box on the computer screen, or provide for the user to utilize one of the function keys to cause the computer to display explanatory information about menu choices if and when desired.

142. And, unless a developer set out to provide precisely the same functionality to the user, in precisely the same way, using a similar menu structure and explanations of equivalent length, there is no logical reason that the contents of whatever explanatory information the program provided should match that communicated in 1-2-3.

* * *

Executed this 27th day of November 1989 at Ann Arbor, Michigan.

_____

Bernard A. Galler

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

SECOND DECLARATION OF BERNARD A. GALLER

I, Bernard A. Galler, declare:

1. I submit this second declaration to supplement my prior declaration dated May 9, 1991, and in support of the Motion for Summary Judgment filed by Lotus Development Corporation ("Lotus") and in opposition to the Cross-Motion for Summary Judgment filed by defendant Borland International, Inc. ("Borland").

2. I have reviewed the Declarations of Professor David Kieras, Dr. David Liddle and Professor Judith Olson submitted on behalf of Borland. I have seen the transcripts of each of their depositions. I have also read a redacted version of the memorandum of law submitted by Borland.

Credentials

3. At the risk of injecting an overly personal note into these proceedings, I was surprised and distressed by the nature of

the attack on my credentials by my junior colleagues from the University of Michigan.

4. I know both Professors Olson and Kieras. I taught Professor Olson her first course in computer science while she was a graduate student here in the late 1960's.

5. While I certainly respect both as serious and hard-working scholars, I do not believe either is in a position to question my expertise. Neither is a computer scientist by training or qualification; their degrees are in psychology. Within the Electrical Engineering and Computer Science ("EECS") Department of which I am a member, both are perceived primarily as applied psychologists.

6. In fact, Professor Kieras has exaggerated the significance of his position within the EECS Department. He is *not* tenured in the EECS Department, contrary to what he states at Paragraph 7 of his Declaration. A request to grant him tenure in the EECS Department some years ago was not approved. The seat of his tenure is in the Program in Technical Communication of the College of Engineering, where he deals with technical writing, a distinct specialty for which he is as well or better known than for user interface study.

7. Similarly, contrary to the implication of Paragraph 1 of his Declaration, Professor Kieras is not "the" user interface specialist within the EECS Department, in the sense that the Department has not appointed or delegated him to cover that subject. He proposed to offer the user interface course he mentions at Paragraph 8 of his Declaration, which was not previously offered; it is an undergraduate course (open to enrollment by graduate students) and is not the only one in the department that addresses user interface issues. I cover such issues, for example, even in my introductory computer science courses.

8. Professor Kieras also mentions at Paragraph 10 of his Declaration that he serves on the dissertation committees of doctoral candidates "working on user interface issues". None

of his three current advisees, however, is a graduate student in computer science.

9. Moreover, I read nothing in the background of either Professor Kieras or Olson to suggest that either has any particular expertise in the development of commercial software products or indeed in microcomputer software development at all.

10. For example, the program Professor Kieras describes at Paragraphs 28 and 44-45 of his Declaration sounds like a batch processor for performing text string comparisons. From what he says in his deposition, its user interface was indeed "very minimal", consisting of one command to run and one command to exit the program. (Vol. 2, p. 333) (Copies of pages from Professor Kieras' deposition referred to herein are attached as Exhibit A.) A program of such limited utility is not much of a basis from which to extract generalizations concerning the role of user interface design in software development.

11. Professor Olson's depth of knowledge concerning computer programming and software development is called into question by two topics covered in her deposition. In one example, she stated her belief that all data sorting routines must involve some form of numerical calculation in their internal workings, based upon her experience in writing a small sorting program which translated letters into numbers and then compared the numbers to determine their order of precedence. (Vol. 2, pp. 55-56 & 82) (Copies of pages from Professor Olson's deposition referred to herein are attached as Exhibit B.) Her comments reflect little understanding of standard programming techniques involved in writing sorting routines, even those that would ordinarily be covered in an elementary programming course or a basic textbook.

12. Her description of the menu commands of Lotus 1-2-3 as a "system for performing numerical calculations" at Paragraph 28 of her Declaration is itself an indication that her understanding of spreadsheet software is incomplete. Quite simply, the menu commands do not perform numerical cal-

culations and, with rare exceptions, are not invoked to perform such calculations. As I was careful to state in my First Declaration in the *Paperback* case (the "*Paperback* Declaration") at Paragraphs 105-106, the menu commands address "spreadsheet-related" tasks or operations, such as the manipulation of data.

13. The other questionable example from Professor Olson's deposition is her conclusion, based upon an experiment in which she observed expert users work with 1-2-3, that the "File" commands are rarely or infrequently invoked. She apparently failed to consider that because the users in her experiment were not given data files with which to work, as would be typical in real usage, her conclusion might have been faulty. (Vol. 2, pp. 129-133)

14. The retrieval and storage of data in memory files is a fundamental computer operation, typically performed throughout a work session. For that very reason, the "File" command in many microcomputer products (such as Microsoft's Excel) is placed in the position in the top-level menu that Professor Olson's own "rules" of user interface design would reserve to the most frequently used command. I find it telling that Professor Olson's common sense did not cause her to question her conclusion that "File" commands are seldom used.

15. Lotus' proprietary research concerning command usage under actual work conditions, which I have reviewed, suggests that "File/Retrieve" is the second most frequently invoked command sequence in the Lotus 1-2-3 menus.

The "Field" of Human-Computer Interaction

16. Human-computer interaction ("HCI") or, as it is alternatively called, computer-human interaction ("CHI"), is a relatively new and broadly-defined interdisciplinary field. The Special Interest Group on CHI of the Association for Computing Machinery ("ACM"), for example, held its first conference in 1982; several journals in which its practitioners, such as Professors Kieras and Olson, publish have sprung up

since then. In contrast, the Special Interest Group on Programming Languages—of which I have been a member since its inception—has been active for more than 25 years.

17. Subsequent to reading the declarations of Borland's witnesses, I made a point to find out more about their background and their publications. I agree with many of the statements concerning the status of this field made by Professor Kieras in his article, "An Overview of Human-Computer Interaction", published in the *Journal of the Washington Academy of Sciences* (June 1990), Vol. 80, No. 2, pp. 39-70 ("Overview of HCI").

18. As he writes, computer science generally does *not* include HCI (*id.* at 42):

> "[C]ourses in HCI or user interface design are not common in computer science departments. Where they do exist, they tend to be viewed with some skepticism . . . . The field of HCI generally has a problem in that it has not established its credibility as a technical discipline with mainstream computer science." (*Id.* at 43)

19. I am a computer scientist. I have never pretended to be a psychologist, "cognitive scientist" or an "expert" in HCI techniques, even assuming that accepted standards exist for such expertise.

20. While the formation of the ACM Special Interest Group on CHI demonstrates a level of interest in the subject among ACM members, it does not confer any formal recognition by the computing profession. I am a former president of the ACM and remain active in its affairs. Special Interest Groups are voluntary associations created at the initiative of like-minded ACM members. They cover any number of subjects, including computing for the handicapped, the social implications of computing, etc. Membership and positions of responsibility within a Special Interest Group are essentially open to those with the time and inclination to pursue them.

21. HCI also has a credibility problem with software development practitioners. The image lingers of the stereotypical efficiency expert armed with the modern version of a stopwatch, measuring the milliseconds it takes for a user to press the keys on a keyboard—the kind of activity Professor Olson describes at Paragraph 29 of her Declaration. It is difficult for most software developers to relate this kind of analysis to their work or self-image.

22. As Professor Kieras has written,

> "Within industry, computer specialists are usually not aware of HCI techniques or results, and often do not take usability [of the user interface] seriously; they believe it is simply a matter of subjective opinion rather than a specifiable design goal like other aspects of the computer hardware and software".

Overview of HCI at 43.

23. Faced with this reaction, it is perhaps unsurprising that Professor Kieras expresses frustration in his efforts to achieve credibility for HCI. Placing himself among the "would-be cognitive engineers" faction of the psychologists' wing of HCI, who believe that "the HCI specialist should be able to do *engineering*, just like the rest of the design team" (emphasis in original), he laments:

> "the problem for this approach is that it has not really been invented yet; there are no convincing demonstrations that the approach is useful in product design. In addition, the engineering approach to HCI seems to rub almost everybody else in the field the wrong way."

Overview of HCI at 45.

24. Although much of the psychological jargon of HCI may be new in the computer field, the issues of user interface design that it attempts to address (in its own way) are scarcely new. User interface issues have existed, in some fashion, for

as long as computing machines have existed. Those of us who were active in computer science from before the time it was known as computer science did not discover the "user interface" when the first psychologist tried to apply his or her experimental techniques to study a person using a computer.

25. The theoretical contributions of HCI may eventually add some insight to the analysis of software design, but they cannot yet substitute for practical experience. As stated in prior declarations, I have participated in writing hundreds of computer programs and have been involved in the design of numerous large-scale systems. Moreover, far from shrinking from my consulting engagements, I am proud of them since they keep me abreast of the concerns of industry and the cutting-edge issues in the field.

26. Professor Olson's false deduction from her spreadsheet usage experiment is an example of the kind of glaring "real world" error that would cause the practitioner to dismiss HCI as irrelevant to the work at hand.

The Bounds of Advocacy

27. I assume that Professors Kieras and Olson believe that the result they are trying to accomplish—to help Borland win and thereby to "free" command hierarchies from proprietary restriction—is right, and that they may believe that the kind of comments about me they have put their name to is all part of the game.

28. I cannot be so charitable towards Borland's lawyers, however, in particular Gary Reback, Edmond Gregorian and Stuart Meyers of Fenwick & West. Late last year I worked with these men as a witness on "clean room" development procedures in a litigation involving their client, Compuware. They knew of my work in the *Paperback* case and discussed it with me in some detail, in the course of preparing and defending my deposition in the *Compuware* matter.

29. Although they might well have anticipated, under the circumstances, that Lotus would call upon me again, not once

in that entire engagement did the Fenwick attorneys mention to me that they were simultaneously representing Borland in this litigation against my client, Lotus. I am still shocked and embarrassed that I was not told.

30. The accusation they make in their brief that I plagiarized from Fred Brooks' book, *The Mythical Man-Month*, is false and out of bounds. Ironically, the trial testimony they refer to in their brief is from a transcript I gave them to assist in defending me in the *Compuware* litigation.

31. Clearly I know of Fred Brooks' work. I have known Fred from long before that book was published and consider him a friend. Certainly I have referred to his work in the past, as do many others in this field.

32. I did not copy from Fred's book and I did not try to pass his work off as my own. There are many "rules of thumb" in software development. Even Professor Kieras, I note, in his deposition described the allocation of one-half of the coding effort to generating the user interface for certain types of programs, which he describes at Paragraph 47 of his Declaration, as "sort of a shared rule of thumb that many of us have used". (Vol. 2, pp. 339-343)

33. My "rule of thumb" is supported by my own experience as a software developer, programmer, designer, manager, consultant and teacher for more than 35 years. Never before in that experience has my personal integrity been attacked.

Points of Contention

34. The vigor of the attack on me is somewhat puzzling in light of the relatively few points in my declarations with which Borland's witnesses actually disagree. Despite the claim that I am supposedly untutored in user interface design, none of the witnesses objects to any of my statements concerning the factors a software developer takes into account in designing an appropriate user interface for a product, or the various alternatives and trade-offs to consider in that design.

35. Instead, both Professor Kieras (at ¶¶ 41-49) and Dr. Liddle (at ¶¶ 38-42) fault what they perceive as an overgeneralization on my part concerning the relative significance of user interface design, as opposed to program coding, in software development. The thrust of their criticism seems to be that, for some types of programs with nominal user interfaces, the coding effort could be relatively more difficult or time consuming than the design.

36. Even assuming this point to be true, I fail to see its pertinence. My discussion assumes, of course, that we are speaking of programs that are designed to have meaningful user interfaces, since I understand user interfaces to be the subject at issue here.

37. In making this criticism, moreover, Professor Kieras is not very careful in distinguishing among the different types of development activity. For example, to illustrate his point about the difficulty of coding, he describes (at Paragraphs 44 and 46) two projects in which the development of the programs' internal workings appeared difficult to him. His discussion imprecisely lumps together the determination of the programs' functionality with the coding effort itself; the former, however, is typically viewed as part of a program's design.

38. Moreover, the determination that an "adequate" user interface would be sufficient for the CAD program he mentions at Paragraph 46 itself reflects a prior design decision. Professor Kieras tells us nothing about the considerations that went into that decision, and he revealed in his deposition that he was involved neither in the design of the user interface of that program nor in its coding. (Vol. 2, pp. 333-339)

39. Professor Kieras also speaks loosely when he refers in Paragraph 48 of his Declaration to designing the "details of the user interface for usability", and asserts that such "details" are typically left to the "coders" to resolve at the end of the development process. Although it is unclear what he means by "details", he cannot mean to suggest that the entire user interface design is nothing more than such a

"detail". If so, his experiences must have involved some very poorly designed software.

40. Dr. Liddle discusses, at Paragraph 40 of his Declaration, the degree to which a user interface is specified in the kind of functional specification document that is frequently drawn up in the preliminary design phase of the development process. The creation of such a functional specification typically marks one specific point in the design process, and the documents vary widely in their level of detail with respect to all elements of a program, depending on the developer's needs, resources and goals. Some are very explicit; some are sketchy.

41. By referring to the functional specification, Dr. Liddle misses my point entirely. I did not state that the user interface is the "most critical task" in *functional specification*, but rather in the "software design and specification" process as a whole. Indeed, in my *Paperback* Declaration at Paragraphs 29 and 30 I distinguished between the functional specification and the "detailed creation of the program's user interface"; I suggested only that the developer should "begin to define" the software's externals, including its user interface, in the functional specification.

42. I do not disagree with Dr. Liddle's point at Paragraphs 46 through 48 of his Declaration concerning the difference in difficulty between developing a "clone" that merely "mimics the behavior" of another product, and one which provides commercially acceptable performance. I made that very point in Paragraph 98 of my *Paperback* Declaration, immediately following the paragraph from which Dr. Liddle quotes.

43. My point about "cloning", as plainly stated in Paragraph 99 of my *Paperback* Declaration, was simply that it is not technically difficult to achieve, holding performance issues aside, and that it is in all events easier than the design and development of original software. I do not read Dr. Liddle as disagreeing with those propositions. Certainly he cannot believe that Borland encountered any great conceptual or

design difficulties in the process of incorporating the 1-2-3 menu tree into its products.

44. Both Dr. Liddle and Professor Olson at Paragraphs 54-55 and 35 of their respective Declarations make what I consider to be an absurd criticism. Both assert that there is a limited range of words that could be sensibly employed to represent the 1-2-3 menu commands in the 1-2-3 menus, assuming that all prior design choices made by the developers remain the same, such as the use of words rather than icons, first-letter keystroke association, and the arrangement of the menu tree.

45. However, the statement in Paragraph 88 of my *Paperback* Declaration to which their comments are addressed was not confined to Lotus 1-2-3, menu-driven user interfaces or even spreadsheet programs specifically. Even for spreadsheet programs, the potential variations in user interfaces or menu trees cannot be analyzed exclusively by reference to the 1-2-3 menu tree. The question should be, how many alternatives to the 1-2-3 menu tree are there for expressing a spreadsheet user interface? The question Dr. Liddle and Professor Olson put is, how many different ways are there to express the 1-2-3 menu tree? The question is invalid and answers itself.

46. Moreover, I fail to see how either could interpret my reference to "the constraints of language" as anything other than a recognition that, when words are employed in a user interface, the "language"—*i.e.*, the words—must communicate to the user. That is not so much a "constraint" on the number of choices available to the developer in designing a user interface as an underlying goal or premise of user interface design.

47. Another instance where Dr. Liddle and Professor Olson disagree with me only because they have made a different implicit assumption is their assertion, at Paragraphs 67 and 50 of their respective Declarations, that it is incorrect to state that a spreadsheet program providing the same functionality as 1-2-3 need not use the same commands.

48. From their preceding discussion, it is clear that both disagree with me only because they adopt a definition of 1-2-3's "functionality" that includes the user's ability to modify and debug 1-2-3 macros by on-screen reference to the 1-2-3 menu tree, while using some other spreadsheet program.

49. Again, both have framed the question improperly. They ask, is there a way to recreate 1-2-3 without using the same menu tree? The valid question is, could a spreadsheet program do everything 1-2-3 does, including execute macros, without duplicating the 1-2-3 menu tree? In my opinion, there is no doubt that it could.

50. Moreover, Dr. Liddle and Professor Olson's point, at bottom, goes to the somewhat different issue of compatibility, not to the feasibility of alternatives to the 1-2-3 menu tree. In my opinion, not only is it possible to provide "macro compatibility" with 1-2-3—*i.e.*, the ability of another program to execute macros written originally for use with 1-2-3—by means of a translation or conversion process, but it also is possible to provide for the modification or debugging of those translated or converted macros by the same process.

51. The set of "Menu Equivalent Commands" in Quattro and Quattro Pro, and their association with both the 1-2-3 and "native" menu trees in those products, illustrates how the "mapping" for such a translation or conversion process could be achieved. They further suggest that those products could have easily provided a conversion capability instead of duplicating the 1-2-3 menu tree in their "compatible" modes.

52. Yet another disagreement without substance is Professor Kieras' erroneous reading of Paragraph 71 of my *Paperback* Declaration (at Paragraph 37 of his). I observed generally that a developer should, in choosing the organization of a menu tree, seek to design one that "assists the user in selecting the commands necessary to accomplish specific tasks". Professor Kieras does not dispute this.

53. Instead, he pretends that I was stating that the words in the 1-2-3 menus provide sufficient information to be a sub-

stitute for the product's documentation, tutorials or help screens—which I have never asserted.

54. I *did* state in my *Paperback* Declaration at Paragraph 118 that the 1-2-3 menus, including the long prompts and their organization, have an "informative dimension like that of a *thumbnail* user's manual, especially for new or infrequent users" (emphasis added), and I mentioned the additional contribution of the "Help" facility in the very next sentence. This is still my opinion.

55. Nevertheless, it seems entirely irrelevant to me whether or not the menus and long prompts taken as a whole, or the menu commands themselves, would necessarily serve as a complete substitute for a user's manual or training course, or could provide *all* the information a user would need to learn how to use 1-2-3. It remains true that the menu commands convey valuable information to the user in that regard. Professor Kieras' point goes only to how *much* information they provide, not *whether* they provide it.

56. Moreover, to the extent he quibbles over a characterization of that information as "explanatory" or "descriptive" or "identifying" in nature, he cannot deny that the act of conveying that information is a form of communication.

57. Professor Olson's criticism of my use of the term "mental model" at Paragraphs 91 to 94 of her Declaration is another fuss over nothing. The term "mental model" is not universally defined and, in any event, I believe my definition at Paragraph 30 of my *Paperback* Declaration is entirely consistent with the definition she gives at Paragraph 93 of hers.

58. Again, she simply misreads Paragraph 106 of my *Paperback* Declaration to claim that I equated the 1-2-3 menu structure with its "mental model". What I said was that the menu structure presents the user with a "particular approach to, or model for, the solution of spreadsheet-related tasks and problems"—in exactly the same sense and using almost exactly the same terms as her lengthy discussion at Paragraphs 58 to 62 of her Declaration concerning alleged differences in the "problem solving approach" or "approach [that 1-2-3 and Quattro Pro] make[ ] available to the user for solving problems".

59. Her concluding remarks at Paragraph 94 about the 1-2-3 menu commands themselves not aiding the user in determining which commands to choose (with which I do *not* agree) and therefore not comprising a "mental model" or metaphor are nonsensical in this context. They have no relation to any point I have made.

60. Borland's witnesses greatly exaggerate the differences between the menu trees in the Quattro and Quattro Pro "compatible" modes and in Lotus 1-2-3. I noted in my prior declaration that there were "extensions" to the 1-2-3 menu tree in those products. I mentioned that the majority of additional commands in Quattro stem from four such extensions.

61. From my review, it appears that most of the additional commands in Quattro Pro are found in similar extensions, located in a few places in the menu tree several levels deep. For example, in Quattro Pro one large group of these commands is located at least four levels into the menu tree (following the "Worksheet/Global/Default/Hardware" command) and pertains just to the task of setting printer specifications, which is a limited area of functionality that users would seldom invoke.

62. Most 1-2-3 users would focus on the higher levels of the menu tree, where the differences were slight. If they did not choose to select one of the additional items there, they would never encounter the great majority of additional commands, which branch from those selections. Thus, 1-2-3 users would perceive the "compatible" modes as virtually identical to 1-2-3, regardless of the numerical count of items deep below.

63. In addition, I fail to see the "difference" in "problem solving approach" Borland's witnesses see between 1-2-3 and the Quattro Pro "compatible" mode (for example, at Paragraphs 58 through 62 of Professor Olson's Declaration).

64. For example, I cannot see the distinction in "approach" between "present[ing] next steps to the user", as Professor Olson says the 1-2-3 long prompts do (at Paragraph 61 of her Declaration), and "explaining the result that will be accomplished" by selecting a command, as she says the Quattro Pro long prompts do. Both speak to what will happen after the user selects that choice, to help the user determine whether that is the choice he wants to make.

65. Since the essential purpose of the menu tree is to help guide the user to the next choice, I do not see how telling the user "where he has been, not where he is headed", which Professor Olson in the same Paragraph says the cascading menus in Quattro Pro do, would be of much help to the user at all. Neither do I understand why a user would ordinarily *want* to "judge[ ] whether he has taken the correct command sequence" when he is making his next selection, or how he would necessarily know that his prior choices were in any sense "correct".

66. In my opinion, there is no support for the proposition that any user would perceive these "differences" as amounting to a different "approach". To the contrary, I believe the two products present substantially similar approaches to the solution of spreadsheet-related tasks.

Is the 1-2-3 menu tree a "system"?

67. I have studied computer "systems" for many years. As Borland's witness, Dr. David Liddle, graciously noted in his deposition (at pp. 47-48) (copies of which are attached as Exhibit C), my early work in systems design and programming achieved a certain prominence.

68. The term "system" is not well-defined. Generally, it refers to a set of related things for performing a method or process.

69. In a loose way of speaking, a "system" could consist of nothing more than an ordered collection or related set. As applied to computer systems, however, the term usually con-

notes some underlying process for accomplishing some particular result.

70. It might be possible to refer to the 1-2-3 program, as a whole, as a kind of problem-solving "system", because it consists of a related set of components which can perform operations for accomplishing various computational results.

71. I would not call the menu tree of 1-2-3 a "system" because, on its own, it does not perform any operations. It is a communication interface that provides information to the user, to enable the user to communicate with the program and cause the *program* to perform the operations the user desires.

72. In my opinion, it is not true that each identifiable part of a "system" is necessarily a "system" (or "subsystem" of the larger "system"). An example I give is that of a library "system", in which the books in the library are component parts. In that example, I would not ordinarily consider the books to be individual "systems".

73. By extending the term "system" to its broadest meaning, any set of things in which a pattern can be discerned would become a "system".

74. If the underlying operations that a "system" can perform could, at the same time, include the "process" of communication, not only the 1-2-3 menu tree but many other things not typically considered as systems, such as the individual books in my previous library example, would become "systems". Systems designers would ordinarily have no reason to make such an analytical stretch.

75. I also do not believe that the 1-2-3 menu tree becomes a "system" when the 1-2-3 macro language is considered. Macros are written by *reference* to, not "use" of, the 1-2-3 menu tree. Macro-writing is an operation performed by the user that is external to the menu tree itself. Thus, the menu tree is no more a "process" for creating macros than a dictionary is a "process" for writing text.

76. Moreover, 1-2-3 could provide, as does Quattro or Quattro Pro, a macro language that did not closely correspond to the words in the 1-2-3 menus. Any programming symbol could technically be used in a macro language to represent any of the items in the 1-2-3 menus. Thus, there is no necessary linkage between the menu tree and the macro language.

77. Alternatively, 1-2-3 could provide no macro facility at all. If the macro language is what turns the menu tree into a "system", it follows that removing the macro facility would make the menu tree no longer a "system". In that situation, the determining factor in defining the "system" would be an external operation, *i.e.*, the writing of macros. This would not be a sensible outcome under systems analysis.

I declare under penalty of perjury that the foregoing is true and correct. Executed on November 1, 1991.

/s/ BERNARD A. GALLER
Bernard A. Galler

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF MARVIN GOLDSCHMITT

I, MARVIN GOLDSCHMITT, hereby declare under penalty of perjury as follows:

A. *Identity of Declarant*

1. I am the "Marv Goldschmidt" [sic] listed in ¶ 73 of the Affidavit of Mitchell D. Kapor filed in the case of *Lotus Development Corporation v. Paperback Software International and Stephenson Software, Ltd.*, Civil Action No. 87-0076-K (the "Kapor Affidavit"), as one of the five persons of the Lotus 1-2-3 "development team" who met with Mitch Kapor "to determine each word choice, location and menu arrangement" of the menu command hierarchy of Lotus 1-2-3.

2. I was awarded a Bachelor of Arts degree in Psychology from the State University of New York in 1974, and a Master

of Arts degree in Psychology from Virginia Commonwealth University in 1979.

B. *Responsibilities at Lotus*

3. I was employed as Director of Product Marketing for Lotus Development Corporation ("Lotus") from June 1982 to May 1983. I was responsible for all phases of the introduction and initial marketing of Lotus 1-2-3, including product positioning, final feature selection, packaging, product testing, development of training programs, pricing, and content of promotional material. In my role as Director of Product Marketing, I worked closely with Mitch Kapor ("Kapor") and Jonathan Sachs ("Sachs") in the design of the menu command hierarchy of Lotus 1-2-3, as shipped.

4. I served as Vice President of Business Development for Lotus from May 1983 to November 1985. I was responsible for development and implementation of strategies for product planning, design and positioning, international business initiatives, and development of marketing channels. During my tenure as Vice President of Business Development, I chaired the Software Protection Committee of the Association of Data Processing Service Organizations (ADAPSO) for one year.

C. *Subsequent Professional Experience*

5. After leaving Lotus, I worked as an independent consultant in the computer industry from November 1985 to March 1987. From April 1987 to October 1987 I served as a Vice President of Prophecy Development Corporation.

6. From October 1987 to February 1988 I served as the Chief Executive Officer of Singha Systems Inc. From November 1990 to March 1991 I served on the Board of Directors of Associative Design Technology (Westboro, Masachusetts). From April 1988 to May 1991 I served on the Board of Directors of Saber Software, Inc. (Cambridge, Massachusetts).

7. I am currently a member of the Boards of Directors of Symbiotics, Inc. (Cambridge, Massachusetts), Clear Logic, Inc. (Boulder, Colorado) and Persoft, Inc. (Madison, Wisconsin), all of which are involved in the development and/or marketing of advanced software and hardware products in the personal computer industry.

8. I maintain an active consulting practice in which I assist companies with strategic market planning for major projects, particularly with respect to basic strategies and tactics for the development and merchandising of products. Most of this type of work has involved personal computer software products. Some of the more recent examples of this work have been for Saber Software, for which I set up international operations over a period of a year and a half, and for Eastman Kodak, with which I have been working for seven months.

9. Since 1988, I have served as a consultant to a number of venture capital firms, helping them to evaluate investment opportunities in high technology products, especially personal computer software products. I have done such work for Frontenac Company of Chicago, Bessemer Venture of Boston and California, and Burr, Egan & Deleage Company of Boston.

D. *The 1-2-3 User Interface as a Whole*

10. Many of the elements of the Lotus 1-2-3 user interface were drawn from two predecessor computer programs written by Kapor known as VisiTrend and VisiPlot, such as the two-line moving cursor menu, the use of full word command names, command selection based on highlighting and pressing the enter key or pressing the first letters of the command name, and long prompts appearing on the status line following highlighting of a command.

11. Sachs worked for more than a year on the design, implementation, testing and debugging of the code for Lotus 1-2-3. Only after the bulk of the code had been written and tested, and the functions the program was to perform were

settled in about August 1982, was the "command hierarchy" portion of the Lotus 1-2-3 user interface fleshed out and refined.

E. *The Command Hierarchy of Lotus 1-2-3*

12. The "command hierarchy" portion of the Lotus 1-2-3 user interface is a system—a particular step-by-step approach—designed by Kapor and Sachs for the solution of numerical problems using a spreadsheet. By "command hierarchy," I mean the particular hierarchical organization of command names in which, as Kapor describes in ¶ 44 of the Kapor Affidavit, "the selection of one command option from the first level menu [leads] in turn to another array of command options on a second level menu (or 'submenu'), and so on, branching out in a sequence of descending menu levels."

13. The command hierarchy of Lotus 1-2-3 was created to provide to the user an effective and systematic approach to the solution of numerical problems. The hierarchical organization of the menus was chosen to enable the user to map out and perform functional steps for accomplishing particular numerical tasks.

14. The command hierarchy was implemented based upon a number of functional rules that Kapor had set. Some of these rules were based upon principles of cognitive psychology. For example, Kapor felt that no more than seven (plus or minus two) commands should be contained in any given menu, because that is the number of items of information that human beings can retain in memory simultaneously.

15. Similarly, commands to be used most frequently should be grouped at the top of the hierarchy; commands displayed within a given menu should appear from left to right in declining order of likely frequency of usage; and commands representing related features should be grouped together on a single level of the hierarchy.

16. Each command of the hierarchy was identified by a single word. The words were chosen to convey to the user in

the most direct and simple way the function of each command.

17. We ensured that each word in a particular menu began with a different letter so that users could invoke the command by typing only the first letter of that command on the keyboard. As a result, experienced users of Lotus 1-2-3 can become so familiar with the first-letter keystrokes for invoking various commands that they begin to think of those commands as keystroke patterns, and their fingers reproduce those patterns out of habit.

18. The command hierarchy of Lotus 1-2-3—the words and their order—was laid out by Kapor, Sachs and myself in a series of meetings during the final few months of the product development process. My role in establishing the words of the command hierarchy and their order was to make sure that the hierarchy was, within the framework of the functional rules established by Kapor, both marketable and useable.

F. *Derivation from VisiCalc*

19. One of our goals in creating the command hierarchy of Lotus 1-2-3 was to enable experienced users of VisiCalc to move from that product to Lotus 1-2-3 with a minimum of difficulty. Accordingly, there were instances in which we made certain command names or portions of the hierarchy the same as VisiCalc. For example, we chose the slash key ("/") as the key for activating the menus, not only because it was a convenient key to use, but because it was consistent with the menu access procedure with which VisiCalc users were already familiar.

20. In other instances, we were forced to depart from the VisiCalc menus for functional reasons. For example, we had to rename VisiCalc's REPLICATE command to COPY in Lotus 1-2-3, because the first letter "R" had already been devoted to the RANGE command in the top level menu.

G. *The Documentation, the Long Prompts and the Help Facility*

21. "Documentation" was prepared to accompany the diskettes containing the Lotus 1-2-3 computer program. The documentation explains the operation of the program and the use of the command hierarchy system. A number of tutorials were included in the documentation to train purchasers how to use the system through illustrative step-by-step procedures.

22. Similarly, the long prompts and the on-line, context-sensitive help facility were added to the Lotus 1-2-3 user interface to explain to the user the functions of the program. The long prompts were added in a line immediately beneath the menu command line in order to explain the functions and options available in the command hierarchy, so as to facilitate acclimation to the program by new users.

23. More particularly, the long prompts were designed: (a) to enable the user to determine the next choices that will become available if the user selects the currently highlighted function displayed on the menu command line, and (b) when the user has reached the "bottom" of a series of submenus in the command hierarchy, to explain the functional result that will take place when the user selects the highlighted function in the command line.

H. *Reasons for the Success of Lotus 1-2-3*

24. In my role as Director of Product Marketing for Lotus 1-2-3, it was my responsibility to conduct research to determine what features potential users wanted in a new spreadsheet product. Such research was of critical importance to the company, because the success of business software was largely determined by its functional features and power.

25. I conducted this research in part by giving numerous demonstrations of Lotus 1-2-3 (indeed, I believe I was the first person ever to formally demonstrate Lotus 1-2-3 to someone outside the company), and by holding many conferences with potential users to determine what they wanted.

For example, I demonstrated a prototype of Lotus 1-2-3 to a group of potential users at a computer store known as Computer Works in Westport, Connecticut on August 30, 1982 to solicit input on what functional features they would like to have in a new spreadsheet that would run on IBM personal computers.

26. In addition, in early September 1982, I formed a group of ten to fifteen persons who agreed to do extensive testing of various early versions of Lotus 1-2-3 up until the time of its first commercial release. These testers gave us detailed feedback on the product and its functioning. Many suggestions were made, both with respect to functional features of the product and its user interface, many of which were incorporated into the commercially released version.

27. I also followed user reactions closely after the product's introduction to determine what features of the product users liked, and areas in which improvement was needed in later releases.

28. Based upon my market research, I determined that what users seemed to want was a new business productivity product incorporating spreadsheet, business graphics and database functions, and that would improve upon many of the perceived limitations of what was then the leading spreadsheet product, VisiCalc. Many users felt that VisiCalc was too slow, and they wanted a spreadsheet that would run on the IBM personal computer. In addition, they wanted a spreadsheet that would allow them to vary the widths of columns, to format numbers in different formats, and to handle spreadsheets of larger size. Many users also desired to have full words in the menus rather than only first letters, as VisiCalc had.

29. In the Fall of 1982, I wrote a memorandum summarizing how the company should position the new product. In that memorandum, I analyzed the features of Lotus 1-2-3 that I believed would cause it to be a success in the marketplace. Those features included all the ones that customers were requesting, as well as several others, such as the two-line "moving cursor" menu, an on-line help facility, an interactive

tutorial, ranges of cells highlighted in inverse video, named ranges, and a macro facility.

30. By providing these functional enhancements to the spreadsheet portion of Lotus 1-2-3, as well as functionally integrating database and graphing capability into one product, which capability VisiCalc did not have at all, we created a product that was functionally superior to VisiCalc and all other spreadsheets on the market at the time. In addition, we created a product that was specifically engineered to take advantage of the power and hardware design of the new IBM personal computers that had recently been introduced to the market. The success of Lotus 1-2-3 was the direct result of these capabilities and enhancements.

31. The command hierarchy per se contributed to the initial success of Lotus 1-2-3 only insofar as it served as a portal to the powerful functionality of the product. The Lotus 1-2-3 command hierarchy was easy to use and felt natural to the user. I believe that it is possible that alternative particular command names and command hierarchies could have been developed which would have felt as natural to the user as that of the actual Lotus 1-2-3, and the product containing that alternative command hierarchy would still have achieved the same level of success.

* * *

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury under the laws of the United States this 24th day of September 1991 at Burlington, Massachusetts.

/s/ MARVIN GOLDSCHMITT
Marvin Goldschmitt

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

DECLARATION OF DAVID E. KIERAS

I, DAVID E. KIERAS, hereby declare under penalty of perjury as follows:

A. *Summary of Experience*

1. I am the person in the Department of Electrical Engineering and Computer Science at the University of Michigan who teaches computer user interface design. To my knowledge, Bernard A. Galler has never taught a course at the University of Michigan devoted to computer user interface or other human/machine interface issues. My academic credentials are set forth below.

2. As more fully described in ¶ 29 below, I have been retained by major computer companies like IBM and Schlumberger to teach courses and consult with respect to the

design of good computer program user interfaces. I have also given these courses in Europe to Schlumberger's overseas operations.

3. As more fully described in ¶¶ 25-28 below, I have written many computer programs in the artificial intelligence language known as "LISP" (List Processing Language), including a complicated program prepared under contract to the Office of Naval Research for automatically detecting problems of comprehensibility in textual materials. I am currently involved in a project sponsored by the Ford Motor Co. to develop a computer-aided design system for creating ergonomic designs for workplace furniture. I am also collaborating with another Professor of Computer Science on the design of user interfaces for advanced computer aided design systems for electronics.

B. *Academic and Other Professional Credentials*

4. I was awarded a Bachelor of Arts degree in Psychology from Rice University in 1969, and a PhD in Experimental Psychology from the University of Michigan in 1974. I was an R.K. Mellon Postdoctoral Fellow at Carnegie-Mellon University from 1974-76.

5. I have published approximately 31 journal and conference papers, 14 book chapters, two book reviews and one book (as co-editor). Of my published papers and book chapters, 18 are on the subject of the design of computer user interfaces and other human/machine interface topics. I have authored or co-authored approximately 35 technical reports distributed under research contracts, 13 of which are on the subject of the design of computer user interfaces and other human/machine interface topics. I will be co-authoring a textbook on user interface design during the next year or so.

6. I have delivered 17 invited lectures and symposia on computer user interface and related topics. A copy of my current curriculum vitae is attached in Exhibit A to this declaration.

C. *My Work at the University of Michigan*

7. I am an Associate Professor with tenure in the Department of Electrical Engineering and Computer Science, and in the Program in Technical Communication (College of Engineering) at the University of Michigan. In addition, I hold appointments as an Associate Professor in Industrial & Operations Engineering Department and in the Department of Psychology at the University of Michigan.

8. Among the courses I teach is an upper division course titled "User Interface Design and Analysis." This course focuses on the theory and practical implementation of good computer user interface design. Students are required to complete four projects in which they apply some of the methods, guidelines and techniques of user interface analysis and design to actual software or other systems.

9. I am frequently asked to referee papers on various computer user interface topics for the following journals: International Journal of Man-Machine Systems and Human-Computer Interaction. I have also served as a reviewer of papers for the Association for Computing Machinery (ACM) Special Interest Group on Computer-Human Interaction (SIGCHI) Program Committee Conference.

10. I regularly serve on the dissertation committees of PhD candidates working on computer user interface issues. To my knowledge, Bernard A. Galler has never served as a dissertation adviser, or a member of the dissertation committee, for any student working in the field of computer-human interaction (CHI).

D. *The CHI Field*

11. I have read the Declaration of Judith Olson submitted in this case (the "Olson Declaration"). I find her description in ¶¶ 8-20 of that declaration of the academic discipline of CHI to be accurate, and I agree with her statement that academics who work in the field of CHI and computer user

interface design would normally be expected to attend conferences devoted to CHI issues and to publish papers in the proceedings of these conferences and in the various relevant scholarly journals.

12. Over the last 8 years, I have attended approximately 8 conferences, symposia and seminars devoted specifically to CHI issues. I cannot recall ever having seen Bernard A. Galler at any of these meetings, nor has he presented any papers at any of the meetings I have attended.

13. I also regularly read articles in the scholarly journals that publish material related to the CHI field. I am unaware of any paper that Mr. Galler has ever published on the subject of CHI, in any of the scholarly journals or otherwise.

14. I am a member of the following professional societies and interest groups devoted fully or in part to the study of CHI issues and computer user interface design: the Cognitive Science Society, the Psychonomics Society, and the ACM Special Interest Group on Computer-Human Interaction (SIGCHI). To my knowledge, Mr. Galler is not a member of any of these societies, nor does he regularly attend or participate in their activities.

15. I do not consider Mr. Galler to be an expert in the CHI field in general, or in the design of computer user interfaces in particular. I have read the declaration submitted by Mr. Galler in this case (the "New Galler Decl.") and his declaration originally submitted in the case of *Lotus Development Corporation v. Paperback Software Int'l*, Civil Action No. 87-0076-K (the "First Galler Decl."). If those declarations had been submitted to me as part of my course on user interface design, they would have been unacceptable because they demonstrate no knowledge of user interface design theory and methodology.

16. Based on Mr. Galler's academic credentials, with which I am familiar, I believe that other academics and practitioners who work in the CHI field also would not consider

Mr. Galler to be an expert in that field. Indeed, Mr. Galler has referred inquiries he has received concerning user interface issues to me.

17. It appears from Mr. Galler's curriculum vitae submitted with his declaration in this case that he has never had any formal training with respect to CHI issues or computer user interface design and analysis. Indeed, to my knowledge, Mr. Galler has never before held himself out as an expert on CHI issues in general or computer user interface issues in particular.

E. *Other Professional Activities in the CHI Field*

18. Each year, SIGCHI, one of the most prominent and important professional interest groups in the CHI field, sponsors a major conference devoted to CHI issues. These conferences regularly draw several thousand participants from around the world. I am serving as a member of the Organizing Committee for the SIGCHI 1992 Basic Research Symposium. I served as a member of the Program Committee for the SIGCHI 1991 conference and the 1986 Conference on Human Factors in Computer Systems, and am currently on the Program Committee for the upcoming 1992 SIGCHI Conference.

19. I am a member of the Governing Board of the proposed ACM SIGCHI User Interface magazine. Since 1989, I have been a member of the ACM SIGCHI Publications Committee and the Curriculum Advisory Panel.

20. Since 1986, I have been a member of the Scientific Advisory Panel for the Basic Job Skills Research Program of the Air Force Human Resources Laboratory. From 1979 to 1982, I served as a member of the Steering Committee of the Society for Computers in Psychology. I was a member of a review panel convened by the University of Maryland to evaluate its Center for Automation Research, which includes the internationally known Human-Computer Interaction Laboratory.

21. Since 1984, I have been a consulting editor for the Journal of Memory and Language. From 1981 to 1984, I served as a consulting editor of Memory and Cognition. I referee articles submitted for publication for the following journals: Memory & Cognition, Journal of Experimental Psychology: Human Learning and Memory, Journal of Memory and Language, Psychological Review, Psychological Bulletin, Journal of Educational Psychology, Quarterly Journal of Experimental Psychology, Cognitive Psychology, Cognitive Science, International Journal of Man-Machine Systems, Human-Computer Interaction, and Cognition and Instruction.

22. I have done extensive review of work proposals for federal granting agencies, including the Air Force Cognitive Science Program and several programs of the National Science Foundation—the Cognitive Processes Directorate, the Research in Science Education Program, the Information Sciences Program, and the History and Philosophy of Science Program.

F. *My Areas of Research and Special Expertise*

23. My scholarly research has been devoted to two principal areas within the CHI field. First, I consider myself to be an expert on the theoretical principles underlying what makes a system useable to a human, and on science-based approaches to the design of user interfaces. In particular, I study how principles and research from cognitive psychology can be applied to make computer systems more "useable"—easier to learn and easier to use. A special focus of my work in this area is on the development of quantitative models for predicting learning time and task execution time based on a task analysis and pre-implementation description of the user interface of the computer system.

24. Second, I consider myself to be an expert on how people learn about systems such as computers. I study how people learn about computers and other equipment from

technical text—in particular, how they learn procedures for using equipment from textual instruction, and how they learn and use "mental models" to assist them in operating equipment, especially from explicit instruction.

G. *Industrial Experience*

25. In addition to my academic work in user interface issues, I have considerable practical experience in the design and implementation of computer programs. In the mid-1960s, I worked part time for two and a half years as a systems programmer on an IBM mainframe at the Baylor University College of Medicine.

26. During graduate school, I wrote the code for a user-programmable real-time system for the IBM 1800 laboratory computer at the Human Performance Center at the University of Michigan. This system was designed to make it easy for experimental psychologists to set up experiments that would process responses from several subjects on five computer terminals simultaneously. I wrote a similar computer program in the late 1970s at the University of Arizona.

27. I have done programming in the field of artificial intelligence, using the computer language known as LISP. In connection with my academic research, I have written a number of programs in LISP for use in cognitive simulation projects—for example, to simulate the modeling by computer of human-computer interaction.

28. One of the most complicated programs I have written in LISP is the "Computerized Comprehensibility System," which I prepared under contract to the Office of Naval Research. This system, which was approximately 9,000 lines of program code and 13,000 lines of database, was designed to assist writers of technical materials, such as military training documents, by automatically detecting problems of comprehensibility in the materials. This system used artificial intelligence techniques to parse and process natural language

into its component parts. The parts were then analyzed using certain expert system rule-based analyses. The rules drew upon theory and results from cognitive psychological research on comprehension.

29. I have been retained as a consultant by major computer companies such as IBM and Schlumberger to teach courses and consult with respect to the design of good computer program user interfaces to their research personnel and product development and management personnel. I have also traveled to Europe to give these courses to Schlumberger's overseas operations. These courses teach a methodology for good user interface design, and afford an opportunity for extensive discussion and evaluation of specific products and other interface problems of the course recipients.

30. I am currently involved in a project at the Center for Ergonomics at the University of Michigan sponsored by the Ford Motor Co. to develop a computer-aided design (CAD) system for creating ergonomic designs for factory workstations. This system is centered around a graphical user interface for specifying and manipulating human postures. I am co-supervising a graduate student who is working to develop and evaluate a highly efficient user interface for this system.

31. Together with another Professor of Computer Science, I am also working on the design of a user interface of an Electronics CAD (E-CAD) program, which has been licensed to a commercial software vendor for commercial development and distribution. This program allows a computer designer to specify timing relationships for a digital circuit by simply drawing standard "timing diagrams" on the screen. We are also collaborating on the design of user interfaces for extremely sophisticated design systems that automatically generate manufacturing specifications for microprocessor-based computer systems.

H. *Familiarity with Spreadsheet Products*

32. I am very familiar with Lotus 1-2-3, with Borland's Quattro and Quattro Pro spreadsheets, and with several other spreadsheet products.

I. *The Nature and Role of the Command Words and Their Order*

33. I have examined Lotus' European Patent Application number 90310897.5, reproduced in Exhibit B to this declaration, seeking a patent on a successor user interface to Lotus 1-2-3, including its "command structure." In the section titled "Background of the Invention" (page 2), Lotus describes generally the nature and function of a "command structure" in a software application program as follows:

> [T]he command structure of the application defines how users initiate commands or choose options and how they respond to the computer's actions in order to manipulate data in the computer.

34. This correctly characterizes, defines and explains the command structure of an application program and its role and use in the implementation of the program. Command structure means, of course, the collection of commands available in the program and their structural, or hierarchical, relationship.

35. The definition and description in the Lotus patent application is accurate as to application programs generally and as to the command hierarchy of Lotus 1-2-3, versions 1, 1A and 2.0, specifically. This Lotus patent application clearly states, and it is also obvious from the vast body of CHI analysis and research, that the command hierarchy of a spreadsheet application program such as Lotus 1-2-3 constitutes a system that defines the steps or procedures users may follow to perform numerical calculations within a spreadsheet by invoking various commands in the hierarchy. The words of the hierarchy identify the choices of functions available to the user in performing those steps or procedures.

36. The Lotus 1-2-3 command system cannot be used without these words. The words are organized and displayed in a hierarchical relationship, because, as is well known from human factors analysis, this is a more economical way of making the functionality of the system available to the user than displaying all the words at once. In addition, the hierarchical organization imposes, as the Lotus patent application notes, a "logical and operational structure" (page 3, col. 4, lines 14-15).

37. I have read ¶ 71 of the First Galler Decl. in which he asserts that the hierarchical organization of commands in menus "assists the user in selecting the commands necessary to accomplish specific tasks." To the extent this statement is meant to suggest that the words of the hierarchy *explain* to the user how to use the system, I believe it is incorrect.

38. As previously noted, one of my special areas of expertise and research is how people learn about computers and the procedures for using them from technical text. The command words of Lotus 1-2-3, although technical words, in no sense explain to the user the procedures for using Lotus 1-2-3. Those words merely identify the functions that are available in the system that the user may use to follow procedures to accomplish a result.

39. To the extent the user of Lotus 1-2-3 needs explanation and education in what the proper procedures are and in making the correct choice of functions to accomplish a task, that explanation is provided by the documentation for the program, by the help screens, by the tutorials Lotus supplies with the program, and to some extent by the long prompts in the second menu line. I have devoted a great deal of my research to studying ways to improve textual instruction of this type to increase the speed and ease with which a user can be taught the proper procedures. It is absolutely clear that the words in the command hierarchy are part of the system, not the explanation of operation of the system. The command words themselves and their hierarchical organization do not teach or explain to the user such procedures.

J. *The Relative Effort of User Interface Design and Coding*

40. I have read the assertions in the First Galler Decl. to the effect that the bulk of effort in developing a computer program is spent in generating the specifications (including the user interface) for the program, and not in the coding of the program.

41. Specifically, Mr. Galler asserts in ¶ 40 that coding "is less difficult and requires less creativity and imagination than developing the design in the first place." In ¶ 37, Mr. Galler concludes that "[g]enerally, coding should not consume more than one-fifth or one-sixth of the total time spent in implementing a new program."

42. Mr. Galler then asserts at ¶ 51 that the user interface is the most important part of the product specification:

> Within software specification and design, expression of the user interface is probably the most critical task the application developer faces and that in which the most significant creative activity and effort is expended.

43. The net impression created by Mr. Galler's assertions is that the user interface is more important to a product than the code that produces both the program's computational results and the screen display itself. This a gross overgeneralization that is simply not accurate, both with respect to the role that user interface specification plays in the design of many products and the code itself.

* * *

81. Giving a monopoly on the command hierarchy of a program requires competitors to make gratuitous changes to the command hierarchy of the competing product, producing a patchwork that would result in significant detriment to users

in the form of inefficiency, retraining and greatly reduced ease of learning and use, and tedious rewriting of macros.

82. From my years of study and research in the field of CHI and human factors analysis, the single most important thing governing the ease with which users learn and use a new computer program is consistency in the user interface across programs. Human factors analysis and research indicate that requiring each new publisher of a type of product to adopt a new command hierarchy breeds inefficiency and deleterious effects on the usability of new products. Hence, from the standpoint of benefit to software users and the advancement of software development generally, there ought to be a strong presumption against a requirement imposed by the legal system that a designer of a new product adopt a new command hierarchy.

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury under the laws of the United States this 26th day of September 1991 at Ann Arbor, Michigan.

/s/ DAVID E. KIERAS
David E. Kieras

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF ROBERT H. KOHN

1. I am Vice President, Corporate Affairs and Secretary of Defendant Borland International, Inc. ("Borland"). I make this declaration in opposition to the motion of Plaintiff Lotus Development Corporation ("Lotus") for summary judgment. I have personal knowledge of the matters set forth below and could testify competently thereto if called upon to do so.

2. I joined Borland as in-house General Counsel on March 19, 1987. Although my title has since changed, I have served continuously as the company's chief legal officer since I was hired.

3. By the time I joined Borland, the software product that became Quattro had been in development for approximately two years. Borland's objective was to design and market an innovative spreadsheet software product with its own unique

user interface and menu command hierarchy. The question whether to include an alternative, 123-compatible menu command hierarchy in Quattro, in order to provide users with the ability to execute and debug their 1-2-3 command macros, did not even arise until after I joined the company.

4. When the question whether to include a 123-compatible command hierarchy in Quattro was first presented to Borland's management in the summer of 1987, we were, of course, aware of the fact that Lotus had sued Paperback and Mosaic for copyright infringement. However, we were also aware of the many public statements by Lotus representatives that the lawsuits that it had filed were directed against "clone" products whose external appearance was virtually identical to 1-2-3.

5. I specifically recall finding in our files at the time this issue arose a copy of the article attached hereto as Exhibit A. The article, which appeared in *InfoWorld* on January 26, 1987, shortly after the *Paperback* and *Mosaic* cases were filed, states that "some Lotus supporters" had become concerned "about where Lotus would draw the line on copyright infringement". The article goes on to quote Lotus spokesman Greg Jarboe to the effect that Lotus' claims had been misinterpreted and that the lawsuits were targeted "at two vendors who had copied 99%" of the 1-2-3 product. Other public statements of Lotus officials repeatedly characterized Lotus' target as "cheap rip-offs or imitations", products which "imitate 1-2-3 as closely as possible" and "1-2-3 clones". (See Exhibit B attached hereto)

6. We were confident that whether it included a 123-compatible command hierarchy or not, Quattro could not possibly be characterized as a clone, much less a 99% clone, of 1-2-3. The dissimilarities between these two products, and between Quattro and VP Planner, are discussed elsewhere in Borland's opposition papers and I will not repeat them here.

7. In connection with our decision whether to include a 123-compatible command hierarchy in Quattro (which would be contained in a discrete program file called the 123.RSC

file), I was directed by our Chairman, Philippe Kahn, to make certain that doing so would not infringe Lotus' copyrights in the 1-2-3 program. Since this was a determination that I was unwilling to make myself, I decided to obtain an opinion from outside counsel. Before we received counsel's advice, we had decided that we would not include the 123.RSC file in Quattro unless we obtained a favorable opinion on the infringement question. We had also decided that we would make any changes to the product that counsel recommended. Because of the manner in which Quattro was designed, such changes could be made quickly and easily.

8. To ensure that Borland would receive frank and totally disinterested advice on the question of infringement, I decided to seek an opinion from a law firm that was not our regular outside counsel and that did not represent us on a day-to-day basis. I retained the firm of Brown & Bain, which I understood to be one of the most well-known and highly respected firms in the country specializing in intellectual property issues relating to computer technology. Brown & Bain's clients include IBM, Intel, Apple Computer, Inc. and other leading computer and semiconductor companies, and the firm has been counsel of record in many of the leading cases involving copyright protection of computer programs, including *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983) and *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q. 2d 1177 (N.D. Cal. 1989).

9. Roger Borovoy, a Brown & Bain partner, was assigned to head up this engagement. Prior to joining Brown & Bain, Mr. Borovoy had been General Counsel of Intel Corporation, a leading supplier of computer chips, including the 8088 line of chips found in virtually every IBM and IBM-compatible personal computer.

10. I instructed Brown & Bain to conduct a thorough factual and legal investigation. Beginning in early to mid-September, 1987 Mr. Borovoy spent several days at Borland interviewing Quattro development personnel and examining the product and documentation. I was purposely not present

at any of the interviews in order to avoid influencing the results of the investigation in any way.

11. After investigating the law and the facts, Mr. Borovoy called and advised me that he had concluded that including the 123.RSC file in Quattro would not constitute an infringement of any Lotus copyright. I requested an opinion letter to that effect and Mr. Borovoy said that he would prepare one. I also reported Mr. Borovoy's conclusion to Mr. Kahn.

12. On October 28, 1987 I received by telecopy Brown & Bain's written opinion that "Lotus should not succeed in any copyright claim it may assert against Borland based on its 1-2-3 copyrights and Borland's Quattro program or manual". A true and correct copy of the opinion letter is attached as Exhibit C. I distributed copies of the letter to Mr. Kahn and other members of Borland's senior management.

13. By this time, the Quattro product had been publicly announced. The announcement, which was made on September 26, 1987, received first page coverage in the trade press. By the end of October, our advertising had begun to appear in the trade press and consumer magazines as well.

14. Based on the opinion letter from Brown & Bain, Borland decided to include the 123.RSC file in Quattro provided that we did not receive an objection from Lotus. Although we were confident that our legal position was correct and believed it unlikely that Lotus would object to Quattro in light of the Brown & Bain opinion, the obvious dissimilarities between the two products and Lotus' own public statements, we also had every expectation that if Lotus did have some objection to Quattro, they would make it known to us. If Lotus did object, either before or after the release of Quattro, we were fully prepared to determine the nature of Lotus' concerns and to consider making changes to the product in order to resolve them. Indeed, we specifically discussed the fact that Quattro's design gave us the flexibility tc make any negotiated changes in the user interface quickly and easily.

15. On or about November 17, 1987, having heard nothing from Lotus, Borland began volume shipments of Quattro to its distributors, dealers and end-user customers.

16. To the best of my knowledge, from that time until this lawsuit was filed in July 1990, Lotus never notified Borland either orally or in writing that it objected in any way to Borland's inclusion of a 123-compatible menu tree in Quattro or Quattro Pro. Moreover, several events that occurred during the intervening years confirmed our belief that our legal position was correct and that Lotus in fact had no objection to our products.

17. At the time Quattro was released, and from time to time thereafter, Borland placed ads for Quattro and Quattro Pro in *Lotus* magazine. *Lotus* magazine was owned and published by Lotus Publishing Corporation, a wholly-owned subsidiary of Lotus. Copies of our advertising, including a four-page gatefold ad for Quattro attached to the front cover of the November 1987 issue of *Lotus* magazine, are attached as Exhibit D.

18. I understand that Lotus may now contend that it was the policy of its wholly-owned subsidiary to accept all competitive advertising, including advertising of products that the parent company believed were infringing. Suffice it to say that to the best of my knowledge, even if such a policy existed, no one at Borland was aware of it. On the contrary, we viewed acceptance of our advertising by *Lotus* magazine as further confirmation that Lotus did not object to the inclusion of the 123-compatible menu tree in Quattro and Quattro Pro.

19. In November 1987, I attended the Comdex trade show in Las Vegas, Nevada. Comdex is the industry's largest trade show. All of the leading personal computer software companies, including Lotus and Borland, rent booths and demonstrate their products to the several hundred thousand people who attend the show each year.

20. Lotus' booth at the November 1987 Comdex was approximately 50 feet away from Borland's booth. During the four days of the show, for at least eight hours each day, Borland representatives were demonstrating Quattro to anyone who visited our booth and answering questions about the product. Although I cannot state from personal knowledge that Lotus representatives actually did so on this occasion, it is common practice for representatives of our competitors to attend demonstrations of our products at trade shows in order to see what their competition is up to.

21. I did have occasion during the show to visit Lotus' booth. While there, I picked up a copy of *Lotus* magazine, containing the four-page gatefold advertisement for Quattro, which Lotus sales personnel were handing out at the Lotus booth.

22. In May 1988, Borland began discussions with Surpass Software Systems concerning possible acquisition af the technology assets of that company. Some months earlier, Surpass had released a 1-2-3 compatible spreadsheet (also called "Surpass") which, unlike Quattro, contained only a 1-2-3 command hierarchy.

23. During the due diligence I conducted for the acquisition, I was informed by Surpass' President that Surpass had never received any notice or claim by Lotus of alleged infringement of Lotus' intellectual property rights.

24. In June 1988, Borland publicly announced its acquisition of the Surpass spreadsheet technology and the transaction was widely reported in the trade press. The transaction closed in September 1988. Borland incorporated the Surpass spreadsheet technology that it had acquired into the product that became Quattro Pro, which was not completed and released until over a year later. At no time during the eighteen-month period between the announcement of the acquisition of the Surpass technology and the shipment of Quattro Pro did Borland receive any notice from Lotus that it objected to Quattro or to the Surpass spreadsheet product.

25. Borland and Lotus were co-defendants in the *Refac* patent infringement suit, involving the 1-2-3, Quattro and Quattro Pro spreadsheets, and had frequent communications and conferences in connection with the case, yet Lotus' counsel never even suggested that Lotus had any objection to the Borland spreadsheet products.

26. Before volume shipments of Quattro Pro commenced in November 1989, I became aware of another commercial spreadsheet product whose user interface was similar to Quattro and Quattro Pro. SuperCalc 5, from Computer Associates, was announced in June 1988 and described in the trade press as offering "menu, macro and file compatibility" with 1-2-3 and "an optional 1-2-3 menu interface". (See Exhibit E attached hereto) SuperCalc 5 was released in February 1989 and, like Quattro and Quattro Pro, contains its own native menu command hierarchy as well as an alternative 1-2-3 compatible command hierarchy. I know of no action against or objection to SuperCalc 5 by Lotus.

27. By the time that Quattro Pro shipped, I had also learned of the official position of the United States Copyright Office that "menu screens and similar functional interfaces consisting of words or brief phrases in a particular format are not registrable." This position statement, by the agency charged with responsibility for copyright registration, was reported in the professional literature (See Exhibit F attached hereto) and served to confirm the advice that we had received two years earlier from Brown & Bain.

28. Numerous other events during this period confirmed our belief that Lotus had no objection to Quattro or Quattro Pro. There were technical discussions between the two companies concerning joint development and mutual product support. (See Declarations of Richard Schwartz and Paul Gross) Lotus also served as a beta test site for Borland computer programming language products. Lotus' Magellan product, which permits the user to run a specified list of applications programs stored on hard disk from within Magellan, includes the ability to run both Quattro and Quattro Pro. (See Exhibit

G attached hereto) Finally, Lotus' own advertising, a sample of which is attached as Exhibit H, identifies "1-2-3 compatible menus" as a key feature of spreadsheet software in comparing 1-2-3, Quattro Pro and other leading spreadsheets.

29. Lotus' attempt in the moving papers to show that Borland expected an objection from Lotus all along is erroneous and highly misleading. For example, Borland's indemnity against a claim by Lotus in our April 1988 agreements with outside developers of Quattro, which I negotiated, was requested by counsel for the developers. I readily agreed to this provision precisely because we were convinced by this time that Lotus had no objection to Quattro. In fact, our own licenses of Quattro to third parties contain broad intellectual property warranties and indemnities, with *no* exclusion of claims by Lotus.

30. Lotus' claim that Borland retained counsel to "monitor the *Paperback* trial" is also false. Borland did not have a lawyer or any other representative present during any part of the trial. Lotus' assertion that I contacted outside counsel because I believed all along that Lotus intended to sue us is also false. I contacted outside counsel in early 1990 when I learned of conflicting press accounts concerning discussion of Quattro Pro at the trial. I had no idea why Quattro Pro would even be mentioned and I asked outside counsel to investigate. I subsequently received from counsel transcript excerpts that referred to Quattro Pro. None of these excerpts indicated that Lotus was claiming that Quattro Pro infringed its copyrights. In fact, an article in *Computerworld* that I read at the time reported that Lotus' counsel had identified both Quattro Pro and Excel as products that provided 1-2-3 compatibility "while sporting a significantly different look and feel." (See Exhibit I attached hereto) Nothing in what I read or heard about the trial caused me to change my belief that Lotus did not object to our products.

31. In sum, by the time that Quattro Pro was released, Lotus' affirmative conduct and failure to object to its predecessor Quattro for more than two years, along with the other

events described above, had only increased our conviction that Lotus had no objection to Quattro or Quattro Pro's use of a 123-compatible command hierarchy. Lotus' conduct and inaction was a substantial factor in our decisions to include such a hierarchy in Quattro, and later in Quattro Pro.

32. In reliance upon Lotus' conduct and inaction, Borland has spent millions of dollars acquiring technology, developing Quattro and Quattro Pro and marketing these products in their present form. Borland spent $2.4 million on the Surpass acquisition alone. Development expenses directly attributable to Quattro and Quattro Pro prior to the filing of this suit amounted to at least $6.0 million. Marketing expenses directly attributable to the two products prior to July 1990 were well in excess of $12.0 million.

33. Had Lotus made its objections known in 1987, I know that Borland would have made every effort to negotiate a mutually acceptable resolution. Prior to the release of Quattro, we had already decided to negotiate with Lotus in the event of an objection and to attempt to resolve any dispute in that manner. A negotiated solution clearly would have been preferable to the uncertainty, time consumption and expense of protracted litigation with Lotus. Even if a negotiated solution had proved impossible, however, we would almost certainly have filed a declaratory judgment action and attempted to secure a prompt determination of our rights rather than risking millions of dollars while awaiting the final outcome of a case over which we had no control.

I declare the foregoing to be true under penalty of perjury under the laws of the State of California, on this 27th day of September, 1991, at Palo Alto, Santa Clara County, California.

/s/ ROBERT H. KOHN
Robert H. Kohn

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,
Defendant.

SUPPLEMENTAL DECLARATION
OF ROBERT H. KOHN

I, Robert H. Kohn, hereby declare:

1. I am Vice President, Corporate Affairs and Secretary of Defendant Borland International, Inc. ("Borland"), defendant herein. I have personal knowledge of the matters set forth below and could testify competently thereto, if called upon to do so. I have previously made a declaration in this case. This declaration is intended to supplement my earlier declaration (dated September 27, 1991), and to bring to the Court's attention events which have occurred since that time.

2. Shortly after this action was filed, Borland began considering alternative ways in which Borland might be able to offer its customers the ability to run macros written using 1-2-3 without displaying the 123-compatible menu com-

mands. Towards that end, in mid-1990 Borland began developing what is now known as Quattro Pro's Key Reader feature.

3. As I understand it, the Key Reader is a macro translator that allows Quattro Pro to read and execute many 1-2-3 macros without displaying 123-compatible menus. The Key Reader portion of the Quattro Pro product contains only the first letters of the 1-2-3 commands, along with an associated command which is understood by the Quattro Pro program, known as the Borland "menu equivalent" command. When Quattro Pro is asked to execute a macro written using 1-2-3, the Key Reader identifies the letters or sequence of letters which make up the macro, translates the macro into the appropriate Borland "menu equivalent" command, and executes that menu equivalent command. The menu equivalent commands are original to Borland.

4. The addition of the Key Reader feature allowed users to execute many (but not all) 1-2-3 macros without the need to leave the Quattro Pro native mode. In addition, and as I explained in my previous declaration, before this action was filed Borland had no notice of Lotus' claims against Quattro and Quattro Pro, and Lotus' Complaint only refers to the user interface displayed when these products are used in their 123-compatible mode (*see* Complaint ¶¶ 17, 19). Although Borland felt that that its display of the 123-compatible interface was legally permissible, Borland is a responsible, public company sensitive both to its shareholders and its customers. Therefore, in the event it was later determined that Borland could not display the 123-compatible interface, Borland wanted to offer users the ability to run most 1-2-3 macros in a manner that Lotus and this Court had previously indicated was clearly noninfringing, as described below.

5. I am familiar with this Court's decision in *Lotus Dev. Corp. v. Paperback Software Intern.*, 740 F. Supp. 37 (D. Mass. 1990). I first read this opinion shortly after it was issued, and have read it numerous times since then. It was my understanding from this Court's opinion that this Court held

that the 1-2-3 user interface, taken as a whole, is copyrightable (and, in that case, had been infringed) and that a translator for 1-2-3 macros was permissible and would not constitute an infringement of any copyright in 1-2-3. In particular, I interpreted as significant this Court's statement, at 740 F. Supp. at 69, that:

> defendants could have instead added a macro conversion capability as the creators of Excel have successfully done (the Microsoft Excel Macro Translation Assistant), and could have provided an on-line help function that would show users the VP-Planner equivalent for 1-2-3 commands. . . . (Lotus itself created a "macro conversion utility" to translate macros among different-language editions of 1-2-3 (e.g., North American, international English, French, German, Italian, Spanish, and Swedish)); Turner Decl. ¶ 10 (Docket No. 309).

6. The Key Reader was first introduced in version 2.0 of Quattro Pro. This version began shipping on about October 17, 1990, and was the first version of Quattro Pro to be introduced after this action began. Because the Key Reader was added to the product late in the development process, it was not initially described in the printed documentation but was fully described (along with approximately 20 other features added late in the development process) in the README text file, distributed with the program. This file is available for display to users during the program's installation process. A copy of a printout of the relevant portion of that README file describing the Key Reader is attached hereto as Exhibit A. All versions of Quattro Pro which have been sold since October, 1990 have included the Key Reader feature.

7. In April, 1991, six months after Borland introduced the Key Reader feature in Quattro Pro, I was informed by counsel that Lotus had explicitly taken the position in discovery in this case that macro execution by means of a translator, such as the Key Reader, would not constitute an infringement of its copyright in 1-2-3. In particular, I read and attached significance to the following statement from Lotus' Response to Borland's Request for Admission No. 9:

> Lotus admits that it does not contend that the inclusion in a spreadsheet program of the ability to execute macros originally written using Lotus 1-2-3 by means of a conversion or translation program, standing alone, would infringe Lotus' copyrights in Lotus 1-2-3 . . .

I was also aware that, and attached particular significance to the fact that, in Lotus' Response to Borland's Request for Admission No. 12, Lotus admitted that its claims against the Quattro products were limited to the 123-compatible interface of those products (which does not contain the Key Reader), and that Lotus did not have any claim against the Quattro Pro native interface (which includes the Key Reader).

8. This Court held a Status Conference in this case on June 18, 1991, which I attended. At that conference Henry Gutman, counsel for Lotus, brought up the subject of Quattro Pro for Windows, a product under development at Borland at that time. He stated that

> [Borland has] not said publicly anything about whether or not their Windows version of the product is going to continue to include the 1-2-3 menus and [sic] alternate user interface. . . I just want to put them on notice now that if they do that, Your Honor, we would consider that an occasion on which we would have to come to the Court for a preliminary injunction.

Transcript of 6/18/91 Status Conference, p. 33. I heard Mr. Gutman make these statements and called them to the attention of my colleagues. I found it of particular significance that, despite the fact that the Key Reader had been in Quattro Pro for several months, Lotus did not indicate that the inclusion of the Key Reader in later-introduced spreadsheet products, such as Quattro Pro for Windows, would give rise to any complaints by Lotus. Based on Mr. Gutman's statements, I concluded that future versions of Quattro Pro, includ-

ing the Windows product, could safely include the Key Reader.

9. Finally, I read the briefs submitted during the second round of briefing on summary judgment in this action and saw that Lotus acknowledged that a macro translation ability in a spreadsheet program is acceptable. Referring to the above-cited portion of this Court's prior decision in *Paperback*, on May 4, 1992 Lotus stated:

> [T]he ability of another program to *execute (i.e.*, "run") such macros, once written [using 1-2-3] . . . as the Court held, could be provided through alternate means such as interpretation, translation or conversion (as then existed in Excel or the foreign language version of Lotus 1-2-3). 740 F. Supp. at 69.

Plaintiff's Response to Borland Memorandum in Support of renewed Motion for Summary Judgment, Dkt. no. 177, 15-16 n.22.

10. There is no question in my mind that, at all relevant times, Lotus was aware of Borland's Key Reader feature. This feature has been available in all versions of Quattro Pro which have shipped since about October 17, 1990, and it was displayed and discussed in Borland's first videotape submission to this Court in support of Borland's cross-motion for summary judgment, filed September 30, 1991 and produced to Lotus. In fact, the Key Reader was described in detail in the November 1, 1991 declaration of Lotus' expert, James Emery, at paragraph 131.

11. It has been my understanding, based on what I have read in this Court's written opinions, that the Court also views Lotus' possible claims against Borland as limited to user interface issues, and that an internal macro translation capability would be legal. For example, in its first written opinion in this case, this Court used Microsoft's product Excel, which implements a macro translator and which displays the 1-2-3 commands on the screen, as an example of a product that is not infringing. *See Lotus Dev. Corp. v. Borland Intern., Inc.*,

788 F. Supp. 78, 81 (D. Mass. 1992). In its most recent opinion in this case, the Court once again stated that the macro translation issue was *not* a part of this case:

> Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes . . . that would have presented a different case from the one now before me.

Lotus Dev. Corp. v. Borland Intern., Inc., 799 F. Supp. 203, 214 (D. Mass. 1992). I read these statements by the Court as confirmation that Lotus' claims could not extend to Borland's Key Reader, and interpreted these statements as guidance to Borland of what would not be considered infringing activities.

12. On July 31, 1992, this Court ruled that the display of Borland's 123-compatible menu command hierarchy constituted an infringement of Lotus' copyrights. Although Borland disagreed with this decision, and even though no injunction had issued, in order to mitigate damages Borland took what it viewed as the appropriate steps for a responsible, public company, and immediately removed its 123-compatible interface from its product, thereby forcing Quattro Pro users to rely exclusively on the Key Reader's capabilities for 1-2-3 macro execution.

13. At that time, the question arose whether the Key Reader feature should also be removed from Quattro Pro. In considering the issues again, I reread the relevant portions of this Court's opinions and Lotus' discovery responses in this case. I made the recommendation not to remove the Key Reader in reliance on this Court's statement that translation was not an issue in the present case, and on the positions Lotus had taken throughout this action, as described above. I relied most directly on the guidance provided by the Court in its July 31 opinion addressing the issue of infringement:

> Use of just the initial letters of the command words (together with the long prompts) or of other symbolic

tokens would have been a sufficient alternate method of implementing the system.

Lotus Dev. Corp. v. Borland Intern., Inc., 799 F. Supp. 203, 218 (D. Mass. 1992).

14. In light of the above statement in the Court's July 31 opinion, Borland at first considered implementing an interface which displayed only the first letters of the 123-compatible commands, along with the long prompts. However, because Lotus had also accused Borland of copying Lotus' long prompts (even though almost all of those long prompts are different), and in an effort to take as conservative an approach as possible, Borland decided only to include the Key Reader (which uses only the "initial letters of the command words" which do not even appear on the screen, and which does not use or display the long prompts at all) in future versions of its products.

15. Quattro Pro for Windows began shipping in mid-September, 1992. For the same reasons previously discussed, it was determined that this product would not display the 123-compatible menus, but would include the Key Reader feature. Borland would replace the 123-compatible interface in its products to afford our customers full macro compatibility if allowed to do so by this Court or a higher Court.

I declare under penalty of perjury that the foregoing is true and correct. Executed this 25th day of January, 1993, at Scotts Valley, California.

/s/ ROBERT H. KOHN
Robert H. Kohn

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,
Defendant.

DECLARATION OF THOMAS M. LEMBERG

I, Thomas M. Lemberg, declare:

1. I am Vice President, General Counsel and Secretary of Lotus Development Corporation ("Lotus") and have been since April 1987. I make this declaration upon my personal knowledge as a duly authorized officer of Lotus.

2. On July 2, 1990, the Board of Directors of Lotus authorized management to instruct its attorneys, O'Sullivan Graev & Karabell of New York and Hale and Dorr of Boston, to file this action for copyright infringement against Borland International, Inc. ("Borland"). (A true copy of the minutes of the board meeting of July 2, 1990, as produced to defendants herein in a form redacted to exclude information protected by the attorney-client privilege, is attached hereto as Exhibit A.)

3. When management of Lotus first became aware of the commercial release of Quattro, Lotus was already before this court as a litigant in both the *Paperback* case and the companion case against Mosaic (*Lotus Development Corp. v. Mosaic Software, Inc.*, Civ. Action No. 87-0074-K).

4. In early 1988, Lotus management considered the possibility of the existence of a legal claim against Borland for the inclusion in Quattro of what appeared to us to be potentially a new form of "clone" of Lotus 1-2-3.

5. While we considered that Lotus might well have a valid claim against Borland, we determined not to assert any such claim at the time. Despite our belief in the merits of Lotus' case against *Paperback*, we concluded that it would be prudent and appropriate to wait until we received a decision in the *Paperback* case before making a final determination as to whether to take any legal action concerning Quattro. We also did not want to delay the resolution of the *Paperback* suit, which we feared might occur if we were to commence an action against Borland and that action would be consolidated in some manner with *Paperback*. Moreover, we could not yet determine whether the commercial threat posed by Quattro was sufficiently genuine to require us to incur the costs, diversion of management time and resources, and risks attendant to another piece of major litigation at that time.

6. At the same time, we were confident that we had made Lotus' position concerning the scope of its proprietary rights in the 1-2-3 user interface and menu structure abundantly clear, both in the *Paperback* lawsuit and in other public statements. An example is attached as Exhibit A hereto, consisting of a copy of a statement by Jim Manzi, Lotus' chief executive officer and chairman, that Lotus provided to *PC Magazine*, a recognized trade industry periodical claiming a broad circulation. The statement appeared in its May 26, 1987 edition, among a series of articles discussing the *Paperback* suit and related issues.

7. We also had reason to believe that Borland, or at least its chairman, Philippe Kahn, was aware of the *Paperback* suit and of the nature of the claims made in it. One of Lotus' employees had observed Mr. Kahn's participation in a public debate launched on the CompuServ electronic mail network following the suit, and had printed out a copy of the various communications. Because of this material, I was aware (as

were others of Lotus' management) of the statements made by Mr. Kahn when Lotus commenced the *Paperback* case expressing apparent support of Adam Osborne and Paperback and hostility to Lotus and its legal position. (A copy of excerpts from the CompuServ material is attached hereto as Exhibit B.)

8. In connection with our decision, we resolved not to make any public comments concerning our legal position vis-a-vis Quattro. I was given the management responsibility to enforce this policy on an ongoing basis. At no time did I authorize anyone within Lotus to make any statements on that subject, to Borland or anyone else, and to my knowledge no one did.

9. After Lotus commenced this action on July 2, 1990, I learned for the first time that Borland had filed an action against Lotus on June 29 (the day after the *Paperback* decision) in the U.S. District Court for the Northern District of California. In that action, Borland alleged that Lotus had made indirect threats to sue Borland in the event that Lotus won the *Paperback* case. (Copies of the complaint, amended complaint, and various affidavits served upon Lotus and submitted by Borland in that action are attached hereto as Exhibit C.)

10. As I testified (by Declaration) in that action, to my knowledge, no such threats were ever made, directly or indirectly; certainly, none were authorized by me or, to my knowledge, by anyone else at Lotus authorized to do so.

11. By the same token, to my knowledge, no one from Lotus has ever told any representative of Borland (or anyone else) that Lotus consented or agreed to the appropriation of the Lotus 1-2-3 user interface, or the use of the 1-2-3 command menus and/or menu tree, in Quattro or Quattro Pro. No such statement has ever been made by me or, to my knowledge, by anyone else at Lotus authorized to do so.

12. Indeed, I am unaware of any communication between the two companies, directly or indirectly, on the subject of the

potentially infringing nature of Quattro and Quattro Pro whatsoever prior to the filing of this suit. If any such communication did occur, I am certain that the Lotus participation in it was unauthorized.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct.

Executed at Cambridge, Massachusetts on May 6, 1991.

/s/ THOMAS M. LEMBERG
Thomas M. Lemberg

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DECLARATION OF DAVID E. LIDDLE

I, DAVID E. LIDDLE, hereby declare under penalty of perjury, as follows:

A. *Identity of Declarant*

1. I am a founder of Metaphor Computer Systems, ("Metaphor") and I have served as its Chairman, President or Chief Executive Officer since 1983. Metaphor is a wholly-owned subsidiary of International Business Machines Corporation ("IBM") engaged in the development and marketing of application and systems software.

2. Throughout my career at Xerox Corp., Metaphor and IBM, I have been deeply involved in the design, development, analysis and commercialization of products exploiting graphical user interfaces ("GUI").

3. I hold a B.S. degree from the University of Michigan in Electrical Engineering. I hold Masters' degrees in Electrical

Engineering and Systems Engineering, and a Ph.D. in Computer Science, from the University of Toledo. I hold in my own name (or, I am co-inventor) of several letters patent pertaining to graphical hardware and software design.

4. Prior to founding Metaphor in 1982, I spent 10 years at the Xerox Corporation—initially as a research scientist and ultimately as Vice President and General Manager of the Office Systems Division. I was responsible for the planning, development and launch of Xerox's pioneering client/server architecture, including the Ethernet Local Area Network and a computer system known as the "Xerox Star."

5. A "user interface" is the totality of means by which the user and the computer or computer program interact. It includes, for example, the visual screen display output on the system's video monitor, input devices such as a keyboard, mouse or other device, and a set of commands. It also includes a "conceptual model," or abstraction selected by the developer of the software which users can relate to tasks they are performing. A "graphical user interface" (GUI) is a user interface the screen display of which has significant and demonstrable graphical or pictorial elements, such as icons, animation, pallets and the like. The Xerox Star that I helped developed was the first commercial computer system employing a GUI.

6. I have been asked on numerous occasions to present papers or make presentations concerning the design, development and analysis of user interfaces to industry conferences and symposia and to other companies. For example, in March and September of 1989 I made presentations at the Seybold Desktop Publishing Seminars. Those presentations were subsequently summarized and published in a paper entitled ***Getting Off The Sandbar***, attached hereto as Exhibit A. Similarly, I was invited to make a tutorial presentation on September 25, 1989 at Agenda '90, which is an exclusive seminar sponsored by Stewart Alsop (publisher of PC Letter) for very senior computer industry executives who meet to dis-

cuss new technology and future directions in the computer industry. That presentation was subsequently summarized and published in a paper entitled ***What Makes a Desktop Different***, attached hereto as Exhibit B.

B. *Importance of This Case*

7. I was aware of and familiar with the products at issue in this case well before my personal involvement in the litigation began. Since becoming involved, I have read the declarations of Mitch Kapor and Bernard A. Galler, filed in this case, and I have become familiar with the allegations and claims for relief set forth in Lotus' complaint.

8. This is not, in my view, a "look and feel" case, as was *Lotus v. Paperback*. This case is not about how a software product *looks*; it is about how a software product *works*, and the extent of the concomitant functional monopoly a publisher secures from the relatively modest act of securing a copyright.

9. There is relatively little preclusive effect to software development generally if a publisher holds a monopoly over the way a product looks. On the other hand, there are significant and widespread consequences if a software publisher can prohibit others from publishing their own software that contains, displays and executes the command hierarchy of the first, even in wholly original code, and particularly, if others are precluded from bringing to market software products that execute programs or databases requiring the command hierarchy of the original product.

10. I have been asked to participate as an expert in other highly publicized litigation involving commercially successful software products. Heretofore, I have always refused. I have agreed to participate as an expert for Borland in this case because I believe that if Lotus is successful in asserting copyright infringement against a product that looks demonstrably different from the Lotus product and that interacts

with the user in a demonstrably different way, but which includes a similar or even the same menu command hierarchy, such a result would inhibit software innovation and development and unnecessarily reduce the productivity of application program users.

C. *User Interface Development at Xerox*

11. Much of my personal experience and reputation with respect to user interface design and commercialization comes from work done by me and by those under my management at Xerox Corporation in the 1970's and early 1980's.

12. Prior to the mid-1970's graphical user interfaces were confined to the largest computer systems and reserved for the most exotic applications—for example, military or other computer-aided design applications. We at Xerox foresaw advancements in computer technology that would, in due course, make more powerful computers readily available to non-experts—for example, to ordinary business people. We therefore set out to amalgamate, analyze and apply known user interface techniques so as to prepare a methodology for the analysis and design of new generations of software programs with user interfaces that could more readily be operated by persons who are not computer experts.

13. In the early 1970's, a great deal of theoretical research and pioneering design work was done at the Palo Alto Research Center (PARC), a research arm of Xerox Corporation, to develop new technology, approaches and models for computer user interfaces. In January of 1975, Xerox founded a new division called the Xerox System Development Division (SDD). The purpose of SDD was to begin to commercialize the ideas and technology that had been developed out of the PARC research. At the time of the founding of SDD, I was a member of the research staff of PARC, and I became the new Manager of Systems Development of SDD.

14. In April of 1976, I hired Charles Irby (now Vice-President of Software Development at Metaphor) to lead a working group at SDD whose mission was to study problems of user interface design for a revolutionary new commercial computer to be marketed by Xerox. Irby's team reported to me and worked under my supervision. I personally attended the early meetings of that group. Much of the fruits of the research at both PARC and SDD was published in scholarly papers, for distribution both inside and outside of Xerox.

15. The PARC work, and later the SDD work, enabled Xerox to produce and enhance a computer known as "Alto." Several application programs were written for the "Alto" that had a user interface that included pop-up menus and windows and employed a "mouse." It was the "Alto" computer that Steve Jobs saw before beginning development of Apple Computer's "LISA" and "Macintosh" computers.

16. Xerox's most widely known product based on the PARC and SDD work was the "Star" (now called "ViewPoint"). The Star was the first product to use "icons" on its screen display. "Icons" in the Star were graphical and pictorial images that represented "objects"—that is, icons expressed complete notions of data and the programs (or processes) associated with them. In addition, the Star was the first product to include animation enabling users to move icons, thereby permitting users to perceive that they could manipulate and organize objects such as documents in a manner reminiscent of the organization of one's desk. These documents could be further organized, in keeping with the desktop metaphor, into "file folders" and "file drawers."

17. To my knowledge, the Star was also the first computer product outside of the typesetting system category to use "what you see is what you get" (abbreviated as "WYSIWYG") correlation between screen display and printout, and was the first product to display compound documents (documents with tables, graphics and text) in WYSIWYG format.

18. Finally, the Star designers implemented what were called "property sheets" at the time. Today, they may be known as pull-down, pop-up, tear-off or other kinds of menus. The Star was the first product in which that sort of display was the fundamental medium for invoking functionality and examining the state of an object.

19. The research at Xerox PARC and SDD spawned a wealth of programs far beyond Xerox products alone—Apple's Macintosh and LISA, Hewlett-Packard's New Wave, Microsoft's Windows, and IBM's Office Vision, to name but a few.

20. Most fundamentally, the PARC and SDD research produced a methodology for user interface design that is based upon a taxonomy or classification for user interface analysis. That taxonomy and methodology continues to be used at Xerox and elsewhere—we use it at Metaphor, for example.

D. *The Taxonomy of UI Analysis and Design*

21. As the Xerox research concluded, and as I have stated on numerous occasions in papers and oral presentations, every user interface has three separable components: the user's conceptual model, the control mechanism or command invocation, and the visuals (or "information display").

22. The user's conceptual model is by far the most important of the three user interface components. Indeed, I have estimated that it may be responsible for 60 to 80 percent of the interface's actual usability.

23. The user's conceptual model is the abstraction selected by the software developer which users can relate to the tasks they are trying to do. By invoking a model, the developer is not trying to make the computer program imitate the object being modeled. Rather, the developer is attempting to give users a metaphor which will aid them in employing the program's functionality.

24. The user's conceptual model that underlies both Lotus 1-2-3 and the Borland products is the "spreadsheet"—a collection of cells in spatial relationships that can contain numbers, formulas and cross-references among themselves, as well as, in the case of macros, commands. There are, of course, other metaphors or models that one might invoke for a program to solve numerical problems and manage numerical relationships. In any event, using the spreadsheet metaphor for this purpose is original to neither Lotus nor Borland. It was first adopted by Dan Bricklin in his VisiCalc product, and it made VisiCalc a successful product in spite of the fact that VisiCalc's functionality was wanting and other components of its user interface were primitive or even nonexistent because of the limited displays and power of the computers on which it was originally designed to run.

25. The second most important component of the user interface is the "command invocation" or "control mechanism": the mechanism that enables the user to extract the functionality built into the software. The control mechanism is a "system," by which I mean a set of actions and results defined in particular relationships to one another.

26. Menu items and keystrokes are part of the control mechanism and were clearly identified as such by the Xerox research published in the mid-1970's. Indeed, because interfaces for business applications at this time consisted almost entirely of menu items and keystrokes, the control mechanism component was originally called "Command Language."

27. Over time, developers began to add additional components to the "control mechanism." For example, the control mechanism of the Lotus 1-2-3 product is a keyboard invoked, two-line moving cursor displaying a menu hierarchy two lines at a time, with the first line active and the second line frequently displaying possible future choices, the specific menu items of which hierarchy consist of full words arranged functionally.

28. The "control mechanisms" of the Borland products are dramatically different from that of the Lotus 1-2-3 product. The control mechanism of Quattro Pro, for example, is a mouse-driven, pull-down menu display which lists current active choices and provides an explanation for the chosen function, and which displays all the various command steps by which the user reached the current place in the command hierarchy, but inactive future choices are not displayed. The control mechanism also depicts certain functions by graphical images such as buttons selectable by the mouse, but the images themselves may or may not be part of the control mechanism. Finally, the control mechanism provides for animation for the manipulation of separate spreadsheet documents through the "View" functions on the top-level menu.

29. Many sophisticated user interfaces have a truly separable visual component—images such as icons, for example, that can be manipulated through animation techniques. This component is less important from the standpoint of usability than the other two, but it may be very important for marketing purposes because, at least in some cases, it establishes the vendor's identity with users.

30. As I have repeatedly stated, I believe that every vendor has the right to develop a product identity in the sense of visual displays. Most "look and feel" cases to date, like, for example, *Broderbund Software v. Unison World*, have related only to the visuals of the product and hence implicate product identity as opposed to usability. I believe that the *Lotus v. Paperback* case was one such case in which Paperback's product, VP Planner, wrongly appropriated the very identity of Lotus 1-2-3 to such a degree that a user could mistakenly think he or she was using Lotus 1-2-3, rather than VP Planner. Borland's Quattro and Quattro Pro products, by contrast, do not at all take the identity of Lotus 1-2-3.

31. The separateness of the visuals from the control mechanism and the relatively lesser importance of the visuals vis-a-vis the control mechanism can be illustrated by reference

to the Apple Macintosh screen display. The screen display of the Macintosh includes icons, animation, shading and the like. (See screen shot, Exh. A.) By invoking a function under the "View" menu at the top of the screen that displays objects "by name" as opposed to "by icons", one can eliminate much of the separable visual expression of the Macintosh user interface (screen shot, Exh. B), and yet the program was designed to function and will continue to function just as well. However, if one eliminates the command words from Lotus 1-2-3, or indeed, from the Macintosh, the program would be useless to the user because the menu commands are the very system that the computer program code implements.

32. Release 2.0 of Lotus 1-2-3 has only the most minimal visual component—for example, the inverse highlighting of command terms as the cursor moves and the on-line help text that can be called up by the user for explanation of the commands. However, I have seen advertising and trade industry publication reports regarding Lotus 1-2-3 for Windows and for the Macintosh and it appears that those products do have a separable visual component.

E. *Software Development for Small Systems*

33. The development of a new software program is a time-consuming and often expensive task. A generalized description of software and user interface development is set forth in a well known book by Frederick Brooks, Jr., entitled ***The Mythical Man-Month***.

34. I have read the Brooks book; indeed, I have it on the bookshelf in my office at Metaphor. At page 20 of his book, Brooks describes the following rule of thumb for scheduling a software development effort:

> For some years I have been successfully using the following rule of thumb for scheduling a software task:

1/3 planning, 1/6 coding, 1/4 component test and early system test, 1/4 system test, all components in hand.

This differs from conventional scheduling in several important ways:

1. The fraction devoted to planning is larger than normal. Even so, it is barely enough to produce a detailed and solid specification, and not enough to include research or exploration of totally new techniques.

35. Brooks' analysis was accurate at the time it was written in the early 1970's for large-scale host-based programming projects with extensive programming staff on large mainframes. Nowadays, however, there are many other techniques for breaking a program's development into horizontal layers beyond those proposed by Brooks.

36. Many, if not most, personal computer (PC) programs are not developed and written using the techniques described by Brooks. Personal computer programs, including Lotus 1-2-3, are usually written by much smaller teams than are characteristic of the host-based programming of prior generations. The dynamics of development in the smaller PC teams are much different than described by Brooks. Similarly, personal computer programs often get divided into parts for development purposes and are developed asynchronously rather than sequentially—that is, the various parts of the program are at different stages of refinement at any given time. The Kapor declarations themselves indicate that Lotus 1-2-3 was not developed using the Brooks model.

37. Hence, the Brooks time estimates of various development tasks and particularly the rule of thumb regarding the ratio of time for code vis-a-vis planning functions does not accurately reflect the development of many, if not most, PC programs.

F. *Relative Importance of Code and UI*

38. Prof. Galler expresses the opinion in his declaration that most of the time spent in developing a software product relates to specifications rather than coding:

> [Coding] is less difficult and requires less creativity and imagination than developing the design in the first place.
>
> First Galler Decl. ¶ 40.

39. Galler then asserts that the user interface is the most important part of the product specification:

> Within software specification and design, expression of the user interface is probably the most critical task the application developer faces and that in which the most significant creative activity and effort is expended.
>
> First Galler Decl. ¶ 51.

40. Prof. Galler's assertion that work on the user interface is the portion of software specification and design "in which the most significant creative activity and effort is expended," is simply inaccurate as a generalization. The functional specification of a software product describes what it does on behalf of the user. If the software is highly interactive, the specification will certainly discuss the user interface. If the product is not highly interactive, the specification may still be enormously detailed, but it will not be concerned with the user interface. A program to control the flow of traffic is a good example of a product with an enormously complicated specification, requiring a great deal of creative work to generate, which does not focus at all on user interface issues.

41. Moreover, Galler's assertion with respect to the importance of user interface specification has little if any applicability to the development of Lotus 1-2-3 as described in Mitch Kapor's declarations. The user's conceptual model for Lotus

1-2-3, by far the most significant portion of the interface, comes from VisiCalc. Mitch Kapor's declarations also indicate that much of the control mechanism (*e.g.*, two-line moving cursor) comes from his own early work at VisiCorp on the VisiTrend product. The additional work actually done on the Lotus 1-2-3 user interface at Lotus was merely the identification and organization of command words, which was done at the very end of the development cycle.

42. Most fundamentally, it is imperative to distinguish the value of "coding" to a product (*i.e.*, the act of writing code) as that term is used by Galler (in ¶ 40) from the value of the finished "code" to the product. Even where, as in the Brooks model, the precise step of "coding" is easier or less time consuming than design and specification, that in no way suggests that the resulting code is unimportant or in any way less valuable than the screen display. Indeed, the screen display (*i.e.*, what appears on the screen) is merely the "output" or result of the execution of a portion of the program's code (*i.e.*, the set of instructions written to accomplish a result).

43. Of course, although the screen display is the output of the program, there is no specific or required relationship between the structure of the code and the order of commands on the screen. The same command hierarchy can be generated on the screen from radically different code structures.

44. Portions of the code other than those producing the screen display provide the product's functionality—in this case, the computational functions. The computational code written by Jonathan Sachs contributed enormously to the success of Lotus 1-2-3. That code was written for a new type of computer hardware and it enabled users to run larger spreadsheets far more rapidly than was permitted by VisiCalc.

G. *"Clones"*

45. For purposes of this declaration, I accept Prof. Galler's definition of a "clone"—"a program that imitates another pro-

gram, mimics its behavior and appears identical to it (or virtually so) in external respects, including the user interface." First Galler Decl. ¶ 91.

46. Galler contends (at ¶ 97) that

> [e]ven good college students given a sophisticated application program such as Lotus 1-2-3, its documentation and an opportunity to observe its operation, could develop a program that mimics the appearance and most aspects of the behavior of the original.

This is inaccurate.

47. There are many examples from my own experience that illustrate the inaccuracy of Galler's assertion. Steve Jobs saw Xerox's Alto, for example, but it still took many engineers at his company months and months of development, undoubtedly consisting of thousands upon thousands of person-hours, to produce a competing product with a similar visual display—Apple's "LISA". In short, although it is relatively easy to generate a particular non-functioning static screen display, it is very difficult to "mimic the behavior" of a sophisticated product with a sophisticated, interactive screen display that provides robust functionality.

48. Moreover, mimicking the visual appearance of a product without providing the same or similar functionality will not produce an economically viable product nor will it threaten the success of the original product. Again, examples abound. Digital Research, Inc. ("DRI") produced a software product known as "GEM" and Atari produced a computer known as "Amiga," both of which had screen displays virtually identical to that of the Apple Macintosh. Yet neither succeeded economically because neither could reproduce the robust functionality and ease of use of the Macintosh, notwithstanding their similar screen appearance.

49. In any event, it is difficult to understand what Prof. Galler's analysis has to do with the Borland products. What

Galler says about the importance of the user interface obscures the fact that the interface of the Borland products is *not* similar to that of Lotus 1-2-3—indeed, the only elements the Lotus and Borland user interfaces have in common is that the commands of the Lotus 1-2-3 command hierarchy appear at various places in the much larger command hierarchy of the 1-2-3 compatible user interfaces of the Borland products.

50. Furthermore, the Borland products are not "clones" of Lotus in any sense, including under Galler's definition. As explained below, their screen appearance and their method of interaction with the user are manifestly different, and their behavior and functionality are far more robust, than Lotus 1-2-3. In no sense could Borland, in the creation of the 1-2-3 compatible mode of Quattro or Quattro Pro accomplish the evil that Galler condemns as "cloning":

> by cloning an existing product a developer can avoid almost all aspects of functional specification, much of the detailed design process, and almost all issues in creation of the user interface.

First Galler Decl. ¶ 93.

H. *The Role the Words Play in the Lotus 1-2-3 System*

51. The command hierarchy of Lotus 1-2-3—the command words and their order—is a system for performing numerical calculations on a spreadsheet. The command words are a central and inseparable part of the Lotus 1-2-3 command system. These words identify the functional choices available to the user, and the system cannot be used without them. The words are required in order for the user to determine in the first instance the command steps that will accomplish the user's goal.

52. The command words do not explain to the user how to use the system—they merely identify the functions that are available in the system. To the extent the user needs explanation and education in using the Lotus 1-2-3 system of commands and in making the correct choice of functions to accomplish a task, that explanation is provided by the documentation for the program, by the help screens, by the tutorials Lotus supplies with the program, and to some extent by the long prompts in the second menu line.

53. Sophisticated users of Lotus 1-2-3 often over time become so familiar with the sequences of command steps necessary to accomplish various tasks that they need no longer rely on the words to select the proper steps. Instead, such sophisticated users come to treat the command steps as purely a sequence of keystrokes. For these users, the necessary sequence of commands (and the correlative keystrokes) are merely an automatic stimulus/response sequence that has been learned from intimate familiarity and repeated use of the words of the Lotus 1-2-3 command system.

I. *The Importance of The Specific Words And Organization*

54. Galler's assertions (in ¶ 88 of his First Galler Decl.) that possible variations to the command words of a user interface "are limited only by the resourcefulness of developers and . . . by the constraints of language" is unreasonable. To be useful, each of the words in the command hierarchy of a spreadsheet product must convey to the user the function it identifies. With respect to each command there are a very narrow range of words that can perform this function—for example "copy," "replicate," or "duplicate" might serve for the same function. Moreover, shorter, more common words are preferred to enhance recognition and conserve space on the screen.

55. The particular word originally chosen for each function in Lotus 1-2-3 was of no particular importance to users so long as it conveyed the particular function. Lotus 1-2-3 would have been just as successful if it used "replicate" instead of "copy," or "duplicate" instead of "copy." Of course, the choice of each word had to be consistent with the design rules

of the product—two words starting with the same first letter could not be present on the same level of the hierarchy.

56. I believe that the specific hierarchical organization of commands adopted by Kapor did not contribute to the success of Lotus 1-2-3. Any functional organization that was as logical and orderly as the one Kapor adopted would have worked just as well, because other aspects of the user interface made the product sufficiently usable to overcome any flaws in the design of the command hierarchy.

J. *The Role and Importance of Macros*

57. Although the specific words in the command hierarchy and their specific organization had no particular significance at the time of the development and the initial release of Lotus 1-2-3, the words and their order acquired great commercial significance later because users embedded that system comprised of those words and their order into their own spreadsheet files in the form of macros.

58. I use the term "macro" as defined on page 166 of the Reference Manual for Lotus 1-2-3 Release 2: "A macro is a set of instructions made up of a sequence of keystrokes and commands that you type into a worksheet as cell entries. After you have entered and named the macro, you can invoke it. Whenever you type the macro name, 1-2-3 reads the instructions and performs the specified tasks."

59. Many frequently used macros written by users consist primarily of stored keystroke sequences representing sequences of commands in the menu command hierarchy. However, macros may also contain commands that do not have equivalents in the menu command hierarchy, such as the "/X" commands, which provide certain advanced macro capabilities like branching and customization of menus.

60. People write and use macros for a number of reasons. As stated on page 166 of the Reference Manual for Lotus 1-2-3 Release 2, "Macros are most useful for:

• Automating frequently used 1-2-3 commands

• Typing the same label many times in a worksheet

•Performing a repetitive procedure that requires a series of sequential commands

• Developing a customized worksheet for someone who is not familiar with 1-2-3"

61. Macros are particularly important in corporate environments, as they enable all members of a relevant group to organize and present data in the same way without error or deviation. Frequently in a business environment, spreadsheets containing macros written by one user are distributed to other users, who must be able to execute the macros contained in the spreadsheets given to them.

K. *The Role the Words and Their Order Play in Macros*

62. The words of the Lotus 1-2-3 command hierarchy and their order play a central role in the writing and execution of macros. As this Court stated in its opinion in the case of *Lotus Development Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 at 65: "Because macros may contain many menu choices, the exact hierarchy—or structure, sequence and organization—of the menu system is a fundamental part of the functionality of the macros."

63. When a user writes a macro containing Lotus 1-2-3 menu commands, the user stores those commands in the macro according to the word and letter designation and hierarchical order defined by the Lotus 1-2-3 hierarchical menu command system. In writing the macro, the user uses the words in the menus to select the functional steps that will accomplish the result the user desires to accomplish and automate, and then embeds those steps in the macro using the command designations defined by those words and in the order dictated by the hierarchical structure of the menus containing the words.

64. The user may also use the words in the menus to debug the macro in the event that it does not run correctly or produces erroneous results. The user may look to the menus, for example, to determine whether a step is missing in the macro or the steps are not in the proper order as defined by the menu hierarchy.

65. The need to debug macros arises frequently. Macros are often written for a spreadsheet configured in a very specific way. With subsequent use, the configuration of the spreadsheet often gets changed. Even small changes to the spreadsheet, such as the addition of a row or column (a new item in a budget or a new period of time, for example) can make the macro fail to accomplish the desired result. The user must then examine the macro in detail, step by step, to figure out how to modify it to make it work correctly in the new situation.

66. Once written, the macro represents a set of steps defined by the user that must be performed in a particular designated way—that is, according to the word and letter designation and hierarchical order of the Lotus 1-2-3 menu command system. Because the user has stored menu commands in the order dictated by the Lotus 1-2-3 menu command system, in order to run the macro to accomplish the user's intended result, the functional steps stored in the macro must be executed in the particular way defined by that system.

67. Thus, Galler is simply incorrect when he states (at ¶ 129 of the First Galler Decl.),

> There is no technical constraint compelling even a spreadsheet program providing the same functionality as 1-2-3 to do so by using the same set of commands . . . .

L. *Dissimilarity of Lotus 1-2-3 and Borland Interfaces*

68. The user interface of Lotus 1-2-3 includes the menu hierarchy (the command words and their order), the type of command invocation system (such as pop-up, pull-down, and moving cursor), the long prompts, the help screens, the input mechanisms (such as keyboard and mouse), and other visual aspects of the screen displays produced by the program (such as highlighting of menu items and scrolling down to parts of the spreadsheet not currently visible on the screen).

69. The user interface of Lotus 1-2-3 is very dissimilar—visually, in its physical interaction with the users and in the approach it makes available to the user for solving problems—from the user interface of Quattro Pro in both its native mode and its 1-2-3 compatible mode.

M. *Visual Differences*

70. Visual differences between Lotus 1-2-3 and Quattro Pro are readily apparent from the screen shots described in and submitted with the Declaration of Laurie Flesher in this case. I have examined the screen shots and I find them to accurately represent the screens containing the various commands depicted therein. I have also examined the Lotus 1-2-3, Quattro Pro, and Quattro products themselves in forming my opinions.

71. The command invocation system is presented visually to the user in very different ways in the two products because the products are based on entirely different control mechanisms. In Lotus 1-2-3, the command invocation system appears as a set of word commands displayed in a two-line moving cursor format, in which the first line presents "active" choices of functions that the user may currently select, and the second line often presents "non-active" choices that will become available if the user selects the highlighted command on the first line.

72. Quattro Pro uses an entirely different control mechanism. In Quattro Pro, menu commands are presented inside pull down windows. Commands are presented not only as words but also as graphical images—icons, buttons, and scroll bars. These graphical images, which are entirely absent in Lotus 1-2-3, appear as part of the visual display throughout operation of the program. This same mechanism is used in all product "modes"—Quattro Pro native, Quattro, user-defined, and, of course, 1-2-3 compatible modes.

N. *Differences in Physical Interaction*

73. Quattro Pro's user interface is also radically different in its physical interaction with the user. The command invocation system of Lotus 1-2-3 is designed so that it can be exercised only through use of the keyboard. Commands are selected either by using the cursor keys to highlight the desired command, then pressing enter, or by typing the first letter of the command on the keyboard.

74. In contrast, Quattro Pro is intended to be used with a mouse, rather than exclusively a keyboard, so different physical actions are done by the user, in conjunction with the different visual methods of display. The user may invoke menu commands in one of two ways with the mouse: (a) by clicking and holding the mouse button on a command word to pull down the menu of subcommands under that command, then "dragging" (sliding) the mouse cursor down the list to the desired command and releasing the button, and (b) by clicking and releasing on the menu to pull its window down, then clicking and releasing on the desired command within the menu. Other functions may be invoked by clicking the mouse on icons, buttons and scroll bars.

O. *Differences in Problem Solving Approach Made Available to the User*

75. One of the most significant differences between the user interfaces of Lotus 1-2-3 and Quattro Pro is the difference in approach each product presents to the user for solving problems. I am informed that Lotus has attached great significance to the fact that the second line of the menu invocation system presents "non-active" commands—in other words, potential choices of next steps and not active functions.

76. The Lotus system is designed in significant part to inform users of what *future* choices will be available to the user for taking the next step after the user selects one of the presently available choices. The long prompts supply this information by listing the subcommands that will become available at the next level of the menu hierarchy if the currently highlighted command term is selected at the present level. Thus, at appropriate points, the Lotus system attempts to aid the user in selecting the proper command sequence by showing the user where that sequence is headed next at a given point in the sequence.

77. Quattro Pro presents a different approach to the user for selecting the proper command sequence. In contrast to Lotus, Quattro Pro does not display the next steps that will become available in the sequence after the current step. Instead, Quattro Pro employs cascading menus that remain on the screen, displaying all *prior* steps to let the user observe the command sequence that lead him to a particular point.

78. The Quattro Pro user judges whether he has taken the correct command sequence by examining where he has been, not where he is headed. This difference in approach presented to the user manifests itself in the long prompts. In contrast to Lotus 1-2-3, the long prompts of Quattro Pro never present next steps to the user. Rather, in every case, they explain the result that will be accomplished if the presently selected command is invoked. It is therefore not surprising that Quattro Pro's long prompts have different words than those of Lotus'—they fulfill an entirely different role.

79. Lotus 1-2-3 and Quattro Pro differ in the approaches they present to the user for solving problems in one other significant respect. Quattro Pro displays "state" information in the menus, which shows the current state or setting of various command choices, such as margin width, to assist the user in determining whether the settings are correct or might need to be changed. "State" information is not automatically displayed as part of the Lotus 1-2-3 command hierarchy, and in order to discover state, the user must invoke the command that affects that state or, in some instances, invoke the "State" command.

P. *Differences in Command Hierarchy*

80. I have read Galler's assertions in ¶¶ 29 and 33 of his Declaration submitted in this case that the "menu trees" of Lotus 1-2-3 and Quattro Pro in 1-2-3 compatible mode exhibit "an extremely close parallelism" and are "substantially similar"—indeed, even "virtually identical." These assertions are incorrect, as a simple examination of the menu trees of the two respective interfaces reveals.

81. I use "menu tree" as defined by Prof. Galler in ¶ 75 of the First Galler Decl.—namely, a visualization of the hierarchical menu structure as a "tree," with dependent commands "branching" from commands at prior levels.

82. I have examined the printouts of the menu trees of Lotus 1-2-3's interface, Quattro Pro's 1-2-3 compatible interface, and Quattro's 1-2-3 compatible interface, described in and submitted with the Declaration of Laurie Flesher in this case.

83. Because of the vast size of the menu trees of Quattro and Quattro Pro, it was not practical to print them in schematic form. Therefore, the menu trees have been printed in columns of a spreadsheet in which columns to the right of another column contain commands "lower" in the tree than the command in the column to the left. I find that these printouts faithfully set forth and compare the menu trees of the respective products they portray, and accurately illustrate the vast differences in the various command structures.

84. The spreadsheet printouts quite readily reveal the vast dissimilarities in the menu trees of the products. The Lotus 1-2-3 menu tree contains 469 commands. The menu tree of Quattro Pro's 1-2-3 compatible interface contains 5215 commands, a ratio of 11 to 1 more commands. Although each of the 469 commands of Lotus 1-2-3 may be found somewhere within the menu tree of Quattro Pro's 1-2-3 compatible interface, such commands are interpersed at all levels throughout the 5215 commands of the menu tree of the 1-2-3 compatible interface.

85. There is not a single level of the hierarchy in which the Quattro Pro 1-2-3 compatible command hierarchy is the same as the Lotus 1-2-3 hierarchy, including the topmost level. Significant differences in the menu trees of the two interfaces exist at all levels.

86. Because of these differences in the command hierarchies, there is no screen comparison of the interface of Lotus 1-2-3 and Quattro Pro's 1-2-3 compatible interface that would reveal identity in visual presentation or even in command names in any screen at any point in the hierarchy.

87. The huge number of new commands in the Quattro Pro menu trees are not merely extra choices that have been tacked on to the Lotus 1-2-3 menu tree. Rather, these new functions in Quattro Pro interact with the other elements of the user interface so as to create an entirely different visual perception and physical response by the user, and a different way in which the user interacts with the spreadsheet to get work done.

88. For example, the "View" menu included in the topmost level of the Quattro Pro 1-2-3 compatible menu hierarchy allows the user to change the basic model of presentation of the program—from a full screen single spreadsheet (which is

all that is available in Lotus 1-2-3) to a presentation in which multiple spreadsheets can be stacked one on top of another like physical spreadsheets laid out on a desktop, or "tiled" together in a tiling pattern, with edges of adjacent spreadsheets touching one another. The commands within the View menu allow the user to change the appearance and physical interaction with the elements on the screen in the following ways:

• To choose between a "stacked" or "tiled" presentation of spreadsheets

• To pick which of multiple open spreadsheets is to be the "active" one (the one that the user is currently manipulating)

• To "zoom" the currently active spreadsheet to full screen size

• To change the size or position of the active spreadsheet on the screen

89. Moreover, the WINDOW subcommands under the WORKSHEET menu allow the user to change the appearance and physical interaction with the elements on the screen in the following ways, among others:

• To remove the column and row borders from the screen

•To condense the spreadsheet display into a "map view" consisting of columns that are one-character wide and that contain, in place of data, codes that indicate the type of data in each cell.

90. Similarly, the DISPLAY MODE command of the Quattro Pro 1-2-3 compatible interface also enables the user to change many fundamental aspects of the visual presentation of the interface. The user can elect to have the information displayed on the screen in either "character" mode (in which only standard textual letters and characters can be displayed in an arrangement of 25 rows by 80 columns) or in "graphics" mode. In graphics mode, Quattro Pro makes the following display changes:

• If the user has a mouse, the cell pointer will be displayed as an arrowhead rather than as a block.

• If a graph is inserted in the spreadsheet, the user can see the actual graph on the screen when function key 10 is pressed (in character mode, inserted graphs are indicated only by highlighting to show where the graph will appear when printed)

• Some menus will be displayed as "galleries"—depiction of options (such as graph types) as graphical representations rather than words

• The elements of the display appear in finer resolution, and the mouse cursor movements are much smoother on the screen.

91. In sum, the new functionality available through Quattro Pro's 1-2-3 compatible interface not only produces a different visual display, it greatly alters the way the user can interact with the program.

Q. *Differences in Quattro*

92. Much of the analysis and conclusions I have described with respect the user interfaces of Quattro Pro also apply to Quattro.

93. Quattro used yet a different command invocation mechanism—"pop up" menus. But, like Quattro Pro, Quattro does not show the user where he is "headed" by displaying the next steps in the sequence of commands that will become available if the currently selected command is invoked. Instead, it presents purely "present" command choice information, displaying at any given time only a single pop up menu showing the presently available "active" commands.

94. Like Quattro Pro, the long prompts of Quattro are purely explanatory of the current active function choices. And, again as in the case of Quattro Pro, Quattro provided other explanatory aspects in its interface, such as help screens

and current state information—all of which are either different from Lotus 1-2-3 (as in the case of help screens) or have no correlative in Lotus 1-2-3 (as in the case of state information).

95. With respect to the menu hierarchy, the menu tree of Quattro's 1-2-3 compatible interface contains 3370 commands, compared to 469 commands in the Lotus 1-2-3 menu tree—a ratio of 7 to 1 more commands. Although each of the 469 commands of Lotus 1-2-3 may be found somewhere within the menu tree of Quattro's 1-2-3 compatible interface, such commands are interspersed at all levels throughout the 3370 commands of the menu tree of the 1-2-3 compatible interface. There is not a single level of the hierarchy in which the Quattro 1-2-3 compatible command hierarchy is the same as the Lotus 1-2-3 hierarchy, including the topmost level. Significant differences in the menu trees of the two interfaces exist at all levels.

R. *The Role of the Native Interface and the 1-2-3 Compatible Interface*

96. It is difficult to imagine that users will find the 1-2-3 compatible interfaces easy to use for anything other than writing, debugging or executing macros. When all the enhanced functionality of the Borland product is added at various points in the 1-2-3 compatible interface, the 1-2-3 compatible interface becomes awkward, difficult and inefficient to use as a system for solving numerical problems.

97. Such awkwardness and inefficiency results principally for two reasons. First, addition of new functions into the existing Lotus 1-2-3 command hierarchy often effectively changes the meaning of existing menus, as illustrated by the additions to the Lotus WORKSHEET menu in the 1-2-3 compatible interface. Second, such additions often cause confusion as to where to look for the functionality residing in previously existing menus, as illustrated by Borland's new VIEW menu.

98. Because Lotus 1-2-3 was designed with two-line moving cursor menus, in which commands are arranged horizontally across the screen, the number of commands that can be placed in any given menu is limited by the width of the screen to about ten. As a result of this constraint, the designers of the Lotus 1-2-3 menus were forced to group commands together under a limited number of top level menus. Accordingly, the WORKSHEET top level menu was created to group together a number of miscellaneous commands that, although very diverse in their functionality, affected the entire worksheet.

99. Because much of the new functionality of Borland's Quattro and Quattro Pro products has no correlative in Lotus 1-2-3, such functionality does not really fit well within any of the Lotus 1-2-3 top level menu categories. Nevertheless, because screen width limitations severely limit the addition of new top level menus, the new functionality had for the most part to be placed somewhere within the existing Lotus 1-2-3 top level menus. In particular, much of the new functionality had to be put in the WORKSHEET menu, because that is the only top level Lotus menu akin to a "Miscellaneous" category—all other top level Lotus 1-2-3 menus relate to very specific types of functionality, such as COPY, MOVE, FILE, PRINT and the like.

100. The addition of the new Borland functionality to the WORKSHEET menu changes substantially the meaning of that menu. Previously, it grouped together diverse functions that all related to entering data into, editing and styling the spreadsheet itself. Many of the new Borland functions that were added to this menu, however, are not of this type—such as UNDO and MACRO commands. Thus, the presence of the new Borland functions in the WORKSHEET menu effectively changes its meaning from a category of "WORKSHEET" functions to a category of "OTHER" functions. The user now is unsure of what types of "other" functions that menu contains, which breeds inefficiency and awkwardness in looking for functions.

101. A second illustration of the awkwardness and inefficiency that results from addition of substantial new functionality to the Lotus 1-2-3 command hierarchy is the new VIEW menu. The VIEW menu commands allow the user to see and manipulate multiple spreadsheets on the screen simultaneously. Lotus 1-2-3 has a WORKSHEET WINDOW menu, which contains commands that allow the user to split a single worksheet into two parts and view the parts simultaneously. Because the WORKSHEET WINDOW menu commands all affect only a single worksheet, that menu did not easily accommodate the new VIEW commands, which affect multiple worksheets.

102. Accordingly, Borland created a new menu titled "VIEW" at the topmost level of the 1-2-3 compatible command hierarchy. Because this new menu is at the topmost level, it is apt to confuse the user as to where to look for the old functionality. In particular, I would expect that many users will be misled to look for the old WORKSHEET WINDOW functions in the VIEW menu, rather than in their original location in the Lotus 1-2-3 hierarchy.

103. In sum, users will find it awkward to access the new functionality of the Borland products, as well as the original Lotus 1-2-3 functionality, through the 1-2-3 compatible interface.

S. *Microsoft Excel*

104. In the First Galler Decl., Prof. Galler devotes 16 full paragraphs (¶¶ 172-187) to a discussion of Microsoft's Excel, asserting that Excel has functionality comparable to that of Lotus 1-2-3 "but with a user interface reflecting a significantly different approach to the spreadsheet" (¶ 172).

105. Most of the distinctions that Galler finds between Lotus 1-2-3 and Excel apply equally to the "native" mode of Borland's products. Thus, both Excel and Quattro Pro use pull-down menus (First Galler Decl. ¶ 175), both have a menu

hierarchy, the "content, structure and organization" of which is dissimilar from Lotus 1-2-3 (*id.* ¶ 176), and the "long prompts" are different in both and fulfill a function different from the long prompts of Lotus 1-2-3 (*id.* ¶¶ 177-78).

106. In the current version of Excel (Version 3.0) the "native mode" of that product continues to use pull-down menus with a command hierarchy different from that of Lotus 1-2-3. However, like Quattro Pro, Excel has adopted a Lotus 1-2-3 compatible mode which the user can invoke. In this compatible mode, the Lotus 1-2-3 command hierarchy is displayed on the screen (when called up through the Help facility by pressing the "/" key) and each command can be executed using the same keystrokes as in Lotus 1-2-3.

107. Hence, to the extent that Galler believes that Excel uses "a very different model or approach to the solution of spreadsheet problems" (*id.* ¶ 188), that approach is now identical to that of Quattro Pro.

T. *Legal Protection for Command Hierarchies*

108. It is my understanding that, generally speaking, patents can be secured in appropriate circumstances to preclude a competitor from making products that users can operate in a manner similar to the product on which patent protection is obtained. I am also informed that copyright protection is relatively easy to secure, while patent protection is, generally speaking, far more difficult to obtain.

109. I have been involved in the computer industry for more than 20 years and I have held positions in software design, product development and management of a number of the industry's leading companies. Based on my experience, I believe that extending copyright protection to the command words and their order in a program like Lotus 1-2-3 will not promote investment, innovation, creativity or competition in the software industry generally or with respect to spreadsheet products specifically. To the contrary, extending easily obtain-

able monopoly-like protection will do nothing more than enable the first company that establishes a widely successful product having a particular set of ordered command words to lock users into that company as the sole source of that type of product.

110. Because a substantial base of users may be tied to the use of a particular system of ordered words, in order to run existing macros or to make use of learned keystrokes to invoke functionality, permitting a company to monopolize a particular order of commands will enable it to effectively shut out competition from other companies that might provide superior products to that same base of users.

111. Generally speaking, protection of the control mechanism of a software product, including its command language, has significant preclusive effects. Competitors must create new control mechanisms and convince users both to learn new ways to operate similar products and to reorganize their existing data to accomodate the new systems. In addition, competitors must convince corporate users to tolerate the inefficiency and potential for error inherent in exposing employees to different systems that are intended to accomplish similar functions.

112. I believe that before a monopoly of this type is conferred on a developer, his or her contribution to furthering the art should be very significant, far more significant than I am told is required to secure a copyright. This is not to say that monopolies on particular control systems should not be granted, but rather that such a monopoly ought to be relatively difficult to obtain—more in line with the patent procedure, as I understand it.

113. Conversely, protecting the separable visual features of a software product has lesser and far different preclusive effects. Protecting the separable visual features of a product allows the developer to create an identifiable product identity, to appeal to potential users through aesthetics and, in many

cases, through ease of use. Such protection precludes competitors from trading off of the identity of another which was acquired through investment in marketing as well as product development, but does not require users to suffer the inefficiencies of gratuitously different competing systems. Such protection is more consistent with the relative ease of securing a copyright, as I understand it.

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury under the laws of the United States this 27th day of September 1991 at Mountain View, California.

/s/ DAVID E. LIDDLE
David E. Liddle

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,
Defendant.

DECLARATION OF JIM P. MANZI

I, Jim P. Manzi, hereby declare:

1. I am the President, Chairman and Chief Executive Officer of Lotus Development Corporation ("Lotus"). I have personal knowledge of the facts set forth herein. I submit this declaration in support of the Motion for Summary Judgment of Lotus and in opposition to the Cross-Motion for Summary Judgment of Borland International, Inc. ("Borland").

2. As I have previously testified at deposition, from the summer of 1979 until May 1983 I was a consultant with McKinsey and Company ("McKinsey") in New York City.

3. In that capacity, I directed a management study for Lotus prior to the introduction of Lotus 1-2-3. I began the assignment in November 1982 and completed it early in February 1983. While conducting that study, I had an office and was regularly at Lotus' Cambridge, Massachusetts offices.

* * *

9. Contrary to his Declaration, he was not responsible for planning and implementing the intial marketing plan for Lotus 1-2-3, which was handled by Mr. Kapor, Chris Morgan and others. (I also seriously doubt that Mr. Raburn actually conducted the 150 product demonstrations he claimed in his Declaration between November 1982 and January 1983—over 50 per month.)

10. In his Declaration, Mr. Raburn mentions a software distribution agreement that he says he "successfully negotiated" on behalf of Lotus with Softsel Corporation, a large software distributor. Although it is true that Mr. Raburn negotiated that agreement, it was not a "successful" transaction from Lotus' perspective. The terms of that agreement were so unfavorable to Lotus that one of my first responsibilities when I joined Lotus in May 1983 was to negotiate a termination of that agreement, which I did.

11. Mr. Raburn's brief tenure at Lotus was marked by conflict with Lotus' founder and then President, Mitchell D. Kapor, and the rest of the staff, including Marv Goldschmitt. In fact, one of my duties while at Lotus as a consultant was to attempt to solve this problem, which arose primarily due to a fundamental disagreement about Mr. Raburn's role at Lotus and his management style. I was not able to resolve these differences adequately, and Mr. Raburn never did become an integral part of the Lotus management team.

12. Due to what were described as "differences in operating philosophy", Mr. Raburn resigned on February 11, 1983, less than one month after Lotus 1-2-3 was first shipped. Based on what I know and observed of Mr. Raburn and of his activities and responsibilities at Lotus, I do not believe Mr. Raburn has any first hand knowledge that would permit him to express the views he does in his Declaration concerning what factors did or did not contribute to the eventual success of Lotus 1-2-3. He simply was not there.

RESPONSE TO THE DECLARATION OF MARV GOLDSCHMITT

13. I have read the Declaration submitted by former Lotus employee Marv Goldschmitt in this lawsuit. I have also reviewed portions of the employment file of Mr. Goldschmitt.

14. According to Lotus' records, Mr. Goldschmitt joined Lotus on August 2, 1982, as a Product Marketing Manager. I understood him to be an acquaintance of Mr. Kapor's. In February 1983, he became the Director of Product Marketing for 1-2-3. He held that position for less than four months, until June 1, 1983, when he became the Vice-President for Business Development, which was essentially a staff position without any direct operating responsibility. At that point he ceased to have any direct responsibility for Lotus 1-2-3. He remained in that position until his resignation from Lotus in November 1985.

15. During his tenure at Lotus, I had an opportunity to work with and observe Mr. Goldschmitt on a regular basis. Eventually, Mr. Goldschmitt reported directly to me in my capacity as President and Chief Operating Officer. In fact, within one month of my joining Lotus in May 1983, Mr. Goldschmitt was having great difficulty performing his duties and I was given the jobs of extricating him from his assignments as quickly as possible—beginning with the marketing of Lotus 1-2-3.

16. I am aware of no studies or surveys at Lotus conducted by Mr. Goldschmitt which analyzed the contribution of the 1-2-3 user interface, and/or its menu commands and structure, to the commercial success of Lotus 1-2-3. The only study he was involved with at the time was one conducted by the Gartner Group and completed in June 1983, which reviewed Lotus 1-2-3 and its competitors and addressed their respective strengths and weaknesses. (Attached hereto as Exhibit A.) That study did not purport to measure why users purchased the product. It did, however, in evaluating Lotus 1-2-3's competitive strengths and weaknesses, give 1-2-3 high marks for ease of use, learning time, design and human factors—all of which relate to the quality of the menus and user interface. After Mr. Goldschmitt became Vice-President for Business Development in June 1983, his responsibilities did not include any activity that would have involved such a study.

17. Contrary to Mr. Goldschmitt's assertions, Lotus' original advertising campaign placed great emphasis on its user interface stating, among other things, that Lotus "1-2-3's spreadsheet is easy to learn and use," and that "You simply choose your task from a 'menu' that's written in English." (Attached hereto as Exhibit B, at 46708, 46714.)

18. During his years at Lotus, I observed that Mr. Goldschmitt had an increasingly poor working relationship with Mr. Kapor, which left him increasingly isolated within Lotus and ultimately resulted in his resignation in November 1985.

19. Contrary to Mr. Goldschmitt's Declaration, I do not recall ever having a discussion (informal or not) with him concerning whether Lotus should institute legal action against Mosaic Software, Inc. or Paperback Software, Inc., or any other software manufacturer for that matter.

20. Based on what I know and observed of Mr. Goldschmitt and of his activities and responsibilities at Lotus, I do not believe Mr. Goldschmitt has any first hand knowledge that would permit him to express the views he does in his Declaration concerning what factors did or did not contribute to the success of Lotus 1-2-3.

21. Over the past nine years I have been intimately involved in the marketing of Lotus 1-2-3, in capacities ranging from consultant to Chairman and CEO. Contrary to the conclusions Messrs. Goldschmitt and Raburn claim to have reached based on their relatively brief and limited associations with Lotus, it is my view, based on my experience, that the 1-2-3 user interface has been a key to the product's extraordinary and enduring popularity.

22. As I testified at my deposition, the intuitiveness, simplicity and elegance of the menu commands and structure have contributed significantly to that success. Millions of users have become familiar with our menus, and we value that familiarity highly. Over the years I have seen ample evidence of the importance to 1-2-3 users of the user interface in their decisions to purchase, and to continue to use, Lotus 1-2-3.

23. During the period when Lotus owned Lotus Magazine, it was published by Lotus Publishing Corporation ("Lotus Publishing"), a wholly-owned subsidiary of Lotus. Although I was ultimately responsible for the overall success of Lotus Publishing as a business enterprise, on an operational and management level the subsidiary functioned independently from the parent company. In fact, we carefully maintained the arm's-length status of Lotus Publishing to preserve both its eligibility for publishers' postal rates and its credibility as an objective source of information. From 1987 through June 1990, management of Lotus did not exercise control over the editorial decisions of Lotus Publishing, or screen or review for approval any advertising that Lotus Magazine carried.

24. Attached hereto as Exhibit C is a letter I received on or about February 3, 1987, from one Julian Lange on the letterhead of Ontio Computer Products Corp.

I declare under penalty of perjury that the foregoing is true and correct.

Executed at Cambridge, Massachusetts this 1st day of November, 1991.

/s/ JIM P. MANZI
Jim P. Manzi

CERTIFICATE OF SERVICE

I hereby certify that a true copy of the foregoing was served upon counsel of record for the defendant herein by overnight courier on November 1, 1991.

/s/ PAUL M. O'CONNER, III
Paul M. O'Conner, III

PATTISHALL, MCAULIFFE & HOFSTETTER
Robert M. Newbury
Craig S. Fochler
33 West Monroe Street
Chicago, Illinois 60603
(312) 641-1500

IRELL & MANELLA
Harvey I. Saferstein
Harold E. Kruth
1800 Avenue of the Stars, Suite 900
Los Angeles, California 90067-4276
(213) 277-1010

RAYMOND C. FITZSIMONS, ESQ.
NCR Corporation
World Headquarters Bldg.
Dayton, Ohio 45479
Attorneys for NCR Corporation and NCR Comten, Inc.

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

No. CV 82-5946 KN (Mcx)

EVERGREEN CONSULTING, INC., a corporation,

Claimant,

—vs.—

NCR COMTEN, INC., a corporation,

Defendants.

INTERNATIONAL BUSINESS
MACHINES CORPORATION, a corporation,

Plaintiff,

—vs.—

NCR COMTEN, INC., a corporation,
and NCR CORPORATION, a corporation.

Defendants.

DECLARATION OF ARTHUR R. MILLER

Arthur R. Miller, declares under penalty of perjury pursuant to 28 U.S.C. § 1746, as follows:

1. I am currently a professor of law at Harvard Law School. I have been a law school professor for twenty-three years, (23), previously holding full time teaching positions at the University of Michigan Law School and the University of Minnesota Law School.

2. During the twenty-three (23) years I have been a law school professor, I have taught courses on copyright, unfair competition and intellectual property at each of the law schools where I have worked. I have also conducted many seminars for judges and lawyers on copyright and intellectual property matters, as well as on civil procedure. At Harvard and at the University of Michigan, I have conducted seminars on computers and the law.

3. I was an active participant in the debates, hearings and negotiations that led to the enactment of the Copyright Act of 1976. I was intimately involved in the legislative process as it related to educational usage of copyrighted materials and the implications of computer and photoduplication technologies on copyright.

4. In 1975, I was appointed by President Ford as a Commissioner on the National Commission On New Technological Uses Of Copyrighted Works ("CONTU"). Together with Justice Stanley H. Fuld and Commissioner E. Gabriel Perle, I served on the Software Subcommittee of CONTU. The Software Subcommittee particularly studied various aspects of the relationship between computer software and copyright and had the primary responsibility for the preparation of that part of the CONTU Final Report dealing with computer programs.

5. In 1983, I was active in the legislative process which developed a bill relating to the protection of semiconductor chips and masks, and which eventuated in the 1984 statute protecting those items. In particular, I appeared as a witness before the subcommittee on patents, copyrights and trademarks of the Senate Judiciary Committee.

6. I am currently serving as the standby arbitrator in the computer-copyright dispute between Apple Computer and Franklin Computer involving the infringement of software. I was named to that position by the parties pursuant to the settlement of the case following the decision by the United States Court of Appeals for the Third Circuit.

7. I have also served as a consultant to several litigants in various copyright matters, and to private institutions on copyright matters. In particular, I have advised on matters relating to the work of CONTU and on questions of copyright protection in general. I make this declaration based on my experience with CONTU and my expertise and experience in the copyright law generally.

8. I have been consulted by attorneys representing NCR Comten, Inc. and NCR Corporation in their litigation against IBM. I have read the Declaration of Arthur J. Levine dated July 9, 1984, which I understand was submitted by NCR Comten, Inc. and NCR Corporation in this action. Mr. Levine was the Executive Director of CONTU and was fully familiar with its activities and intent. Based on his experience with

copyright law, and my contact with him at CONTU, I believe he is a recognized expert on copyright law. I agree with the statements made by Mr. Levine in his Declaration of July 9, 1984.

9. In considering the appropriateness of copyright protection for computer software as a member of CONTU, I was particularly concerned, as were most of the other members of the Commission, that copyright should not block anyone from developing their own software, even if that software were to perform the same function as pre-existing software and were to accomplish it in the same way. There was a great concern about not allowing the copyright monopoly to be used to inhibit people from developing the art of programming or to prevent the maximum utilization of information technology for the widest range of uses and users.

10. The entire purpose of my questioning of Daniel D. McCracken at a CONTU hearing was to be sure that programmers were not seeking protection at a level that would damage the social utility of programs or of technology. Thus, I asked Mr. McCracken whether he sought protection of "the intellectual juice" from the program, and Mr. McCracken responded that he would like to have that degree of protection, but that the second comer in that instance would not be taking the expression because of the amount of work required. He likened the program to a play, indicating the dialogue is what should be protected by copyright. The following discussion involving myself, Commissioner Nimmer and Mr. McCracken occurred in CONTU session 10:

> COMMISSIONER MILLER: Could I now circle you back. When we were talking about protection, you listed the example of Chinese copying and use. But given the cost of production of one of these major programs, do you or don't you think that it is necessary to protect against not the literal photocopyist, but the person who sits down either with the source or the object listing,

studies it and says, 'Aha, I see how this person with these people invested a million dollars and came up with ideas 1 through N, and now I am going to go out and produce my own program to achieve the same result with some twist or wiggles and bumps and valleys, but basically having gotten all the intellectual juice out of that million dollar investment.

MR. MCCRACKEN: I would very much like to respond to that. That is an important question. As you may have detected by now, I am a bit of a moralist I don't like thievery, and I would prefer protection against that, too, but in practice that is not really a very big problem, because that second guy has a lot of work to do. He is almost going to deserve what he gets. If you will permit me to tell a very short story. Moss Hart in his autobiography, 'Act One', told about the stupidest thing he ever heard anybody saying about writing a play. He ran into a young playwright at a cocktail party who was known to be working on a play and asked him how it was going. He said, 'Well, I am almost done. All I have to do now is dialogue it.' He said it was the stupidest thing ever because the dialogue is the essence of the play. Until you have done that you do not have a play. You have the idea of a play.

John Hersey said here before, it is an unusual author that can get along on an outline. Some people can, but not very many. If you have the idea of a program, even some very clever idea, you still have a long, long way to go before that is a running reliable product. The original coding, writing the source codes down on sheets of paper, or whatever, is no more than ten or twenty percent of the total cost of writing a program. The rest of it is in designing, how it is going to work and the checking it out, establishing that it works, getting the errors out. So that second guy, picking this up and saying, 'I see how a compiler works,' or something, in the first place he has

a lot of work to do just to understand it, and then he sits down and writes his own, he may spend nearly as much as the original.

Compilers aren't new. In a sense everybody does them. Somebody else sits down to write another compiler from scratch, having read the source code from this, he would still spend about the same amount of money.

COMMISSIONER NIMMER: In focusing on trying to find what is the idea of the program which is unprotected, if we look at conventional copy records the expression may be short of the literal Chinese copy. Take a motion picture based upon a novel, it would be ridiculous to say that because you have motion picture rights on the novel that the work is done. But nevertheless, the one who makes a motion picture based upon a novel without the consent of the novelist is infringing the novel, because you are taking more than the idea. The mere fact that additional work is needed is not necessarily the answer if we look at conventional copyright principles. Maybe in this context we should think that the protection only goes to the literal protection.

11. The members of CONTU did not specifically adopt Professor Nimmer's "literal protection" or "Chinese copy" limitation, but did consider that limitation to be one of the basic goals to copyright protection of software. In listing the "variety of ways" a computer program can be misappropriated in the CONTU Final Report, all the ways involved verbatim copying. The examples given include photocopying, recording of the program on tape and copying the program into the memory of the computer. If anything beyond these uses were to be considered infringement, that determination was to be made by the courts on a case-by-case basis.

12. Ultimately, CONTU adopted the principle that copyright should not be used to block the development of new pro-

grams, specifically stating that as one of the objectives of the proposed legislation. The Final Report stated the objectives at p. 12:

> To provide reasonable protection for proprietors without unduly burdening users of programs and the general public, the following statements concerning program copyright ought to be true:
>
> 1. Copyright should proscribe the unauthorized copying of these works.
>
> 2. Copyright should in no way inhibit the rightful use of these works.
>
> 3. Copyright should not block the development and dissemination of these works.
>
> 4. Copyright should not grant anyone more economic power than is necessary to achieve the incentive to create.

13. To illustrate CONTU's concern that copyright not be used to block the development of other software, or the utilization of the technology, an analogy can be drawn to the experience of two people seeing a baseball game for the first time, and having to describe that baseball game. Because some of the same things must happen in every baseball game, and in the same sequence, the structure of what two people write in describing baseball will necessarily have similarity. For example, before the game starts, the National Anthem is played. The home team is always on the field at the start of the game, and the pitcher must throw to each batter in turn. The definition of what is a hit, a home run, a foul ball or an out does not alter and will be the same for anyone. Similarly, after three outs are made, the teams change sides with the visiting team going to the field and the home team coming to bat. This time, play again begins with the pitcher for the visiting team throwing to the plate. To phrase it simply, describing such a game is a constrained process, with the constraints

being introduced by the nature of the subject matter involved. Copyright should not be used by the first writer of the rules of baseball—its program if you will—to block other authors in writing a second description of the game. And most assuredly, a copyright in the description of baseball cannot be permitted to inhibit anyone's playing the game—practicing the technology of baseball, as it were.

14. The members of CONTU were also concerned that copyright protection for software not create a patent-like monopoly in the guise of copyright. The danger of such a monopoly arising was perceived as acute because of possible antipathy to one person using someone else's ideas. It was, however, strongly felt that this type of protection, which various commissioners occassionally referred to as "anti rip-off," was not appropriate for copyright. The members of CONTU believed that if ideas were to be protected, that protection would have to be under the patent law or trade secret law.

15. In addition to considering patent and copyright protection for software, many CONTU debates focussed on whether there should be protection against misappropriation of a program, that is the taking of the essence or intellectual idea of the program. The conclusion reached was that any such misappropriation protection would have to be separate from the Copyright Act, which protected only the expression of the program. Although the design or logic of the program might represent a major part of the intellectual effort or value of the program, that part is not protected by copyright.

16. I have read the Declarations of Melville B. Nimmer and John A. Baumgarten, submitted by IBM in this action. These Declarations need some amplification or clarification to present more completely the views of the members of CONTU on protection of computer software. Some specific clarification and amplification is also needed so that the statements of Professor Nimmer and Mr. Baumgarten not be misunderstood. Generally, in Paragraph 8 through 13 of this Declaration I have tried to amplify and show the concerns of

the members of CONTU in recommending copyright as a way to protect the expression of computer programs.

17. In particular, Professor Nimmer and Mr. Baumgarten repeatedly state that the scope of protection for computer programs is the same generally as that for other categories of literary works, and specifically list novels, plays and directories as other examples of literary work. That statement, however, only begins the analysis of what is the scope of protection afforded by copyright. Stated simply, the different types of literary works, including novels, plays and directories, do not have the same scope of protection. It has long been recognized that literary works of fiction enjoy a broad scope of protection, often extending to matters such as theme and plot, while nonfiction works have a very narrow range of protection, often limited to verbatim copying or very close paraphrasing. Facts, themselves, are not subject to copyright. One of the most recent examples of this distinction in the scope of protection between fiction and nonfiction literary works is shown in *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F. 2d 475 (9th Cir. 1984). The court there considered a book setting forth a strategy for playing Scrabble Crossword Game. The rules or suggestions for better play were, of course, nonfiction. The Court held the game-playing strategy could be protected only against verbatim copying or very close paraphrasing. In my opinion, a computer program is certainly a nonfiction work in the same sense that a strategy for playing the Scrabble game or the rules for playing baseball is a nonfiction work. The games and the software programs are all constrained by the required rules.

18. I have also considered the statements in the Declarations of Professor Nimmer and Mr. Baumgarten respecting copyright protection for names and short phrases. Although they both recognize that names and short phrases are not ordinarily copyrightable subject matter, the claim is made that if these names or short phrases are included in a compilation they are protectible, and the decision by Judge Learned Hand

in *Reiss v. National Quotation Bureau*, 276 Fed. 717, 719 (S.D.N.Y. 1921), is cited as support for that position. I certainly agree that compilations may be protectable, but clarification is needed if the statements are to be properly considered. Although names and short phrases in a compilation may be protectible as part of the compilation, what copyright protects in the compilation is the collection and assembling of pre-existing materials or of data that is selected, coordinated and arranged. Copyright does not protect individual names or short phrases simply because they are included in a compilation. That would amount to the kind of blockage prohibited by 17 U.S.C. § 102(b) and feared by the members of CONTU.

19. I do not believe that a computer program is a compilation, so the principles respecting protection for compilations would have no application. A computer program does not collect and assemble pre-existing materials. If particular names and short phrases could be protected simply because they appear in a computer program, the same principle would certainly apply to fiction and you would reach the ridiculous stage of having the copyright on *Gone With The Wind* preclude others from using the words "gone," "wind," "frankly" and "damn" because they are part of the "compilation of words" called *Gone With The Wind*.

20. The *Reiss* case cited by Professor Nimmer and Mr. Baumgarten simply dealt with the issue of whether a code book consisting of 6,325 coined five-letter words was copyrightable. Nowhere did Judge Hand suggest that any individual word or name is copyrightable nor did he make any finding as to the scope of protection for those coined words. The *Reiss* case was cited in the CONTU Final Report for two separate purposes, namely: (1) to show the broad interpretation the courts have given to the term "writings" in the Constitution; and (2) for the proposition that copyrightable works do not necessarily have to communicate with human beings.

It was not cited to indicate that protection was to be extended to names or short phrases.

21. The reason, aside from lack of creativity, that names and short phrases are not accorded protection is very simple. To return to my baseball analogy, the round tapered log which is used as a striking tool in baseball is called a "bat;" the fabric covered spheroid that is struck by the striking weapon is called a "baseball;" and the square pillow shaped devices that rest at three points on the playing field are called "bases." For anybody to write about the game of baseball and inform the reader of what he is saying he must necessarily use the words "ball," "bat" and "base" as well as dozens of other non-appropriatable elements, to describe what is taking place. Thus, even if those names were original and creative, they would not be a proper subject of copyright protection. Similarly, names used in a computer program are not a proper subject of copyright protection.

Executed at Cambridge, MA this 3d day of January, 1985.

/s/ ARTHUR R. MILLER
Arthur R. Miller

IRELL & MANELLA
Harvey I. Saferstein
Kenneth A. Liebman
Steven A. Marenberg
1800 Avenue of the Stars, Suite 900
Los Angeles, California 90067-4276
(213) 277-1010

PATTISHALL, MCAULIFFE & HOFSTETTER
Robert M. Newbury
Craig S. Fochler
33 West Monroe Street
Chicago, Illinois 60603

WEIL, GOTSHAL & MANGES
Salem M. Katsh
Robert F. Brodegaard
767 Fifth Avenue
New York, New York 10153

RAYMOND C. FITZSIMONS, ESQ.
NCR Corporation
World Headquarters Bldg.
Dayton, Ohio 45479
Attorneys for Defendants

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

No. CV 82-5946 KN (Mcx)
June 24, 1985

EVERGREEN CONSULTING, INC., a corporation,

Claimant,

—vs.—

NCR COMTEN, INC., a corporation,
and NCR CORPORATION, a corporation,

Defendants.

INTERNATIONAL BUSINESS
MACHINES CORPORATION, a corporation,

Plaintiff,

vs.

NCR COMTEN, INC., a corporation,
and NCR CORPORATION, a corporation,

Defendants.

SECOND DECLARATION OF ARTHUR R. MILLER
(IN SUPPORT OF NCR COMTEN'S MOTION FOR
SUMMARY JUDGMENT AS TO COUNT I
(INFRINGEMENT OF COPYRIGHTED PROGRAMS)
OF IBM'S FIRST AMENDED COMPLAINT)

I, Arthur R. Miller, declare upon penalty of perjury, pursuant to 28 U.S.C. §1746, as follows:

1. I have been asked to supplement my first declaration in this proceeding, dated January 3, 1985, in order to consider the significance of the Patent law's treatment of computer program logic to the question of whether the use of that logic based upon a study of the program's language (e.g., source code) is a matter that Congress intended to be prohibited by the Copyright Act. In my opinion, this question must be answered in the negative.

2. In considering this issue, it is important to recognize the very distinct spheres of responsibility of the Copyright and Patent laws insofar as computer "programs" are concerned.

3. It is important to do so because the operational elements of a computer can bridge both the Copyright and Patent laws, but in critically different ways. As computer operations are technically complex and in a sense "invisible" to those not conversant in computer science (lawyers and judges (and law professors) included) it is easy to confuse the application of the copyright and patent laws to computers, particularly if we use such terms as "computer program" and "expression" in loose or inexact ways.

A. *The Scope of Copyright Protection for "Computer Programs"*

4. Turning first to copyright, the term "computer program" was defined by Congress in accordance with its technical meaning in computer science — "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. §101. This was the definition recommended to Congress by CONTU.[1]

5. A "computer program" as defined in §101 is protected by copyright under §102(a)(1) as a "literary work." Elements of computer operations that do not fall within the definition of "computer program" set forth in §101 will not be subject to copyright protection, as a literary work, under §102(a)(1).

6. Copyright may subsist in matters such as flow charts or logic diagrams that depict computer operations to the extent that such matters may qualify as graphic works under §102(a)(5) of the Act. Flow charts and logic diagrams, however, do not fit the definition of "computer program" in §101

---

[1] As set forth in my first decalaration, I was a member of the National Commission on New Technological Uses of Copyrighted Works ("CONTU"), and served on the software committee.

and hence cannot qualify for protection as a literary work under §102(a)(1).

7. Congress' care in limiting the copyright protection to be afforded to "computer programs" solely to machine-executable programming expression is reflected quite clearly in §102(b) of the Copyright Act, which specifically provides:

> "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, *regardless of the form in which it is described, explained, illustrated, or embodied in such work.*" (Emphasis supplied.)

8. In part, this provision was enacted to deal with possible efforts to extend copyright coverage for "computer programs" to the utilitarian processes underlying them. As stated in both the House and Senate Reports on §102(b):

> "Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than *merely to the 'writing' 'expressing his ideas*. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that *the actual process or methods embodied in the program are not within the scope of the copyright law.*"

S. Rep. 473, 94th Cong., 2d Sess. 54 (1976); H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976) (emphasis supplied).

9. In my first declaration, I pointed out that CONTU felt strongly that the underlying ideas of computer programs should not be subject to patent-type protection under the copyright laws and that the existence of a copyright on a computer program should not in any way inhibit the utilization of computers to achieve the same results that can be achieved

with a copyrighted program. CONTU was urged, for example, to modify §102(b) so as *not* to exclude from copyright protection the logical choices made by those personnel working on the design phase of program development.[2] CONTU rejected this proposal because it would have been inconsistent with the traditional nature of copyright protection for particular forms of expression and not for underlying processes.

10. CONTU likewise declined to adopt a recommendation by proponents of copyright protection for the logic of computer operations that would have amended the Copyright Act to provide that "[a] computer program may be a derivative work of a flow chart and either may be a derivative work of a literary work."[3] Such a provision, had it been adopted, could have effectively granted monopolies over logical processes through the registration of copyrights on flow charts, irrespective of whether such processes would meet patentability requirements.

B. *The Scope of Patent Protection for the "Art" Expressed by Computer Programs*

11. Turning now to the Patent law, it is immediately clear that it does not cover a "computer program" as that term is defined in the Copyright Act. That is, a patent cannot be obtained on the programmer's choice of *programming language* — the form of expression comprising the copyrightable subject matter.

12. What the Patent law does cover is, as set forth in §101 of the Patent Code: "any new and useful process, machine,

---

[2] *See* Transcript of Proceedings CONTU, May 7, 1976, Testimony on behalf of Information Industries Association, at 130-132. The suggested amendment to §102(b) provided as follows:

> "However, copyright protection may exist in a collection of ideas or abstractions, arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes."

[3] *Id.* at 122

manufacture, or composition of matter, or any new and useful improvement thereof. . . ." 35 U.S.C. §101. It is provided in 35 U.S.C. §100(b) that "[t]he term 'process' means process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material."

13. There is an obvious equivalence between the types of matter protectable by the patent laws, and the types of matter *excluded* from copyright protection under §102(b) of the Copyright Act.

14. The Supreme Court has confirmed recently that the processes implemented by computer programs may constitute patentable subject matter. *See Diamond v. Diehr*, 450 U.S. 175 (1981). If these processes reflect sufficient novelty and utility, they are subject to protection under the Patent law, with the patentee having the exclusive right to make, use and sell the invention for 17 years.

15. The Patent law's protection of processes meeting this standard for patentable subject matter is strong evidence of the appropriate demarcation between copyright and patent law in terms of the protection of different elements of computer operations.

16. As set forth in the declaration of James B. Gambrell, the logical routines that a computer program causes a computer to perform — the "results" that the statements or instructions "bring about" — are regarded as processes or methods under the Patent law. *See* Gambrell Declaration ¶ 7.

17. It is plain, therefore, that due regard to the careful terminology used by Congress in defining a computer program in §101 of the Copyright Act, and in excluding "processes" and "methods of operations" from copyright protection under §102(b), requires denial of any copyright claim for protection, as a computer program/literary work under §102(a)(1), of the logical routines the program causes the computer to perform.

C. *Congress' Demarcation of Responsibility as Between Copyright and Patent Law for Protecting Different Elements of Computer Technology is Grounded on the Long-Standing Distinction Between Forms of Expression, On the One Hand, and Systems, Methods and Processes, On the Other*

18. Congress' specific limitation of copyright protection for a computer program is consistent with the utilitarian nature of computer operations. Computer programs are not works of fiction, musical compositions, or other types of works having an artistic or entertainment end purpose. The end purpose of a computer program is to achieve a utilitarian result, *i.e.*, the computer's performance of logical operations in a way that produces the desired practical consequence. One cannot compare, therefore, the underlying processes of a computer program with, say, the underlying plot structure of a novel or a screenplay of a movie. This, of course, is the distinction recognized by the Supreme Court long ago in the seminal decision of *Baker* v. *Selden*, 101 U.S. 99 (1879).

19. The Copyright Act has never protected such things as rules, methods, processes, or systems, even before the statutory codification of that principle in § 102(b). *See, e.g., Morrise* v. *Proctor & Gamble Co.*, 379 F.2d 675 (1st Cir. 1967); *Affiliated Enterprises, Inc.* v. *Gruber*, 86 F.2d 958 (1st Cir. 1936); *Brief English Systems, Inc.* v. *Owen*, 48 F.2d 555 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931). Copyright in terms of utilitarian works subsists solely in those particular *forms of expression* that are recognized by the Act, such as a writing which expresses a process in an original way (§102(a)(1)), or a graphic presentation thereof (§102(a)(5)). These provisions and the cited cases make clear that the subsistence of copyright does not prohibit a user from practicing the art or process described in the work. Copyright cannot be used to block access to the ideas or utilitarian functions described in a copyrighted work.

20. An understanding of Patent law, on the other hand, makes clear that these utilitarian aspects of computer software are not left without protection. Consider the following example. Suppose Steel Company A puts together a team of engineers to create a better process for fabricating steel. Much effort and creative talent go into the work of successfully designing a new process. The engineering team then reduces the process to diagrams and the diagrams are given to other employees with the responsibility of generating a narrative pamphlet explaining the process for use in the company's many plants. The pamphlet is then copyrighted.

21. Surely, it can be seen that the copyrighting of the pamphlet does not protect the utilitarian process itself, no matter how detailed the description of the process within the pamphlet.

22. Thus, if Steel Company B were to obtain a copy of the pamphlet and extract the "logical flow" of each step in the process and write its own manual describing that process in language that does not substantially duplicate the language of the copyrighted manual, no copyright infringement would result even though both pamphlets describe the very same process in equivalent detail. The same result obtains as a matter of Copyright law if Steel Company B actually practices the new steel fabricating process developed by Steel Company A.

23. This illustration is but a direct application of *Baker* v. *Selden*, 101 U.S. 99, 104 (1879), in which a unanimous Court laid down what is undisputedly the law today:

> "The use of the art is a totally different thing from a publication of the book explaining it . . . . Whether the art might or might not have been patented, is a question which is not before us. It was not patented, and is open and free to the use of the public."

24. Quite clearly, what Steel Company A should have done was seek patent protection for the processes themselves. If the processes are not sufficiently novel for patent protection or otherwise do not meet the historic substantive requirements for patent protection, the steel company's complaint should not be directed toward some alleged insufficiency in the Copyright Act, but rather to the public policy choices made by Congress in implementing the constitutional copyright and patent grants.

25. In my opinion, precisely the same infirmity resides in IBM's claim that the copyright it obtained on its program also protects the art — the processes — practiced by the computer when so instructed by the program.

26. This country has for a long time drawn a very definite boundary between the competing interests of encouraging competition through promoting a free market and a constantly expanding public domain in ideas (including utilitarian processes and methods) and rewarding true innovation by means of laws granting limited monopolies over defined subject matters.

27. This is the express purpose of Art. I, §8, c1.8 of the Constitution ("To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries"). As a general rule, public policy is strongly in favor of everyone having full access to and use of the various elements described in §102(b) of the Copyright Act, on the theory that society will be benefited more by the competition and broadened base for experimentation resulting from repeated replications of new ideas and processes than it will be from protecting the first iteration of an idea or process within a legal envelope of monopoly protection.

28. In my view, therefore, the issue of whether a computer's logical processes are part of the copyright subsisting in the "program" (as the term is defined in §101) does not

depend on whether the person devising such operations has "expressed" himself in some semantic sense. Alexander Graham Bell undoubtedly engaged in creative "expression" when he mapped out the first telephone. But had he not secured a patent on that invention, his description of it in a copyrighted book or article clearly would not have given him patent-type protection for his innovative work. Again, this principle was settled long ago by the Supreme Court in *Baker* v. *Selden*, involving a copyrighted book showing a highly detailed bookkeeping system. The Court's famous and still fully appropriate distinction between the realms of copyright and patent seems particularly apt here:

> "[T]he principle is the same in all [cases]. The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent." 101 U.S. at 105.

29. It is thus not the purpose of the Copyright law, as it is the purpose of the Patent law, to provide incentives for industrial or technological progress. Rather, copyright protects only particular forms of "original" expression. The creativity, ideas and utilitarian aspects of a copyrighted work must look elsewhere for legal protection.

30. To argue that copyright protection should be afforded to computer program processes that are not novel enough to satisfy the statutory requirements for patent protection because they are the result of creative effort and the investment of substantial resources, is to voice what may be a legitimate concern in the wrong legal environment. The Copyright law is not the forum for determining how to deal with such a problem, assuming one exists. The history of copyright in the United States and the language of the current statute make clear that copyright is not the right vehicle for IBM to obtain what it wants.

31. The language of the Copyright Act makes it clear that Congress has been unwilling to grant monopolies to utilitarian processes under the aegis of copyright. In the context of computer programs the statutory protection extends only to the statements and instructions that make the machine perform. Consequently, protection for the underlying logic of a computer program must be sought under the Patent Act (or perhaps trade secret law) or not at all.

Executed this 20th day of May, 1985, at Cambridge, Massachusetts.

/s/ ARTHUR R. MILLER
Arthur R. Miller

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

DECLARATION OF JUDITH OLSON

I, JUDITH OLSON, hereby declare under penalty of perjury as follows:

A. *Academic and Other Professional Credentials*

1. I am a full Professor in the School of Business Administration and in the Department of Psychology at the University of Michigan.

2. I was awarded a Bachelor of Arts degree in Mathematics and Psychology from Northwestern University in 1965, and a PhD in Mathematical Psychology from the University of Michigan in 1969. I was a Postdoctoral Fellow in Cognitive Psychology at Stanford University in 1970.

3. I have published approximately 33 papers, 18 of which are on the subject of the design of computer user interfaces and other human/machine interface topics. I have delivered

lectures and papers at over 20 symposia on computer user interface topics. A copy of my current curriculum vitae is attached in Exhibit A to this declaration.

4. From 1982-89, I was a member of the National Research Council's Committee on Human Factors. This Committee is a standing committee whose members are appointed by the National Academy of Science's research arm, the National Research Council. Its members consist of twelve nationally recognized researchers who convene about four times a year in Washington, D.C. to recommend to governmental funding agencies those research areas that should take top priority for funding in the near future. This Committee recommends policy for the National Science Foundation, the Air Force Office of Scientific Research, the Army Research Institute, the Office of Naval Research, the Nuclear Regulatory Commission, and the National Air and Space Administration. During my tenure on the Committee, I participated in or chaired five workshops on human factors issues sponsored by the Committee.

B. *Industrial Experience*

5. From 1980 to 1983, I worked as a member of the technical staff and a technical supervisor at Bell Laboratories in Holmdel, New Jersey. During my tenure there, I designed the user interfaces for new communications computer programs. These user interfaces consisted of a mixture of both graphics and command words.

6. When I first arrived at my Division of Bell Laboratories, I was the only engineer out of a group of 300 engineers devoted to human factors analysis and other human/machine interface issues. During my three years at Bell Laboratories, I built two human factors groups consisting of eleven PhDs in Psychology and Computer Science and eight support persons. In addition, I designed, funded and built a human factors laboratory, implemented a summer internship program for graduate students in psychology and computer science, and helped

shape the current practice of involving human factors analysts through the entire software design process, from market analysis through implementation.

7. I have also served as a consultant to Chrysler Corporation, Unisys, TRW, NYNEX and Electronic Data Systems. This work focused, among other things, on developing methods for efficiently redesigning user interfaces for Computer Aided Design (CAD) systems, expert systems for automobile diagnosis, and banking applications. This work also focused on developing management plans for incorporating human factors analysis into the product development lifecycle.

C. *The Academic Discipline of Computer-Human Interaction*

8. There is a discipline devoted to the study of computer-human interaction (CHI), which includes as a subspecialty the design of efficient and suitable user interfaces for computers. Computers present a special challenge for interface designers, because computers place many cognitive demands on the user—the user must figure out what the computer can do as well as the steps necessary to accomplish desired tasks, and then must be able to remember the proper steps at appropriate times in the course of work. Thus, computer user interfaces must be engineered to be efficient with respect to the cognitive processes users go through in using the computer.

9. The CHI discipline is an increasingly important one as personal computer technology becomes more and more ubiquitous in both business and homes. Efficient, well-engineered user interfaces are a crucial key to realizing the potential of the personal computer to revolutionize the efficiency of business. The CHI discipline is recognized, studied and practiced both in academia and by industrial and business practitioners.

10. Several professional societies and interest groups are devoted to the study of computer-human interaction and computer user interface design. One of the most prominent and

important ones is the Special Interest Group on Computer-Human Interaction (SIGCHI) of the Association for Computing Machinery (ACM).

11. Each year, SIGCHI sponsors a major conference devoted to CHI issues. These conferences regularly draw several thousand participants from around the world. Papers presented at each year's SIGCHI conference are published in the Proceedings of the conference.

12. I served as the technical co-chair of the 1991 SIGCHI conference held on April 28–May 2, 1991, which was attended by approximately 2500 people. In that role, I was responsible for the planning and selection of the technical content of the proceedings and the papers to be presented. I have chaired many other symposia on CHI issues.

13. Several other conferences devoted to CHI issues are regularly held. Examples of these include the Institute of Electrical & Electronics Engineers (IEEE) Workshop on Visual Languages, the ACM Symposium on User Interface Software and Technology, and the American Association for Artificial Intelligence (AAAI) Workshop on Architectures for Intelligent Interfaces.

14. In addition to the ACM, several other societies are devoted to addressing CHI issues, in whole or in part. Examples of these societies include the Human Factors Society and the Cognitive Science Society, of which I am a member.

15. A host of professional journals are devoted exclusively to, or regularly publish, scholarly research on CHI issues. Some examples of these journals include the International Journal of Man-Machine Systems, Human Computer Interaction, Organizational Computing, ACM Transactions on Office Information Systems, IEEE Computer Graphics and Applications, Cognitive Science, Cognitive Psychology, and Management Information Systems Quarterly.

16. I currently serve, or have served in the past, on the editorial boards of the following journals: Management Information Systems Quarterly, Organizational Computing, Human Computer Interaction, Psychological Review, and Memory and Cognition.

17. Academics who work in the field of CHI and computer user interface design would normally be expected to attend conferences devoted to CHI issues and to publish papers in the proceedings of these conferences and in the previously mentioned scholarly journals.

18. Over the last 10 years, I have attended approximately 15 conferences, symposia and seminars devoted to CHI issues. I cannot recall ever having seen Bernard A. Galler at any of these meetings or that he presented any papers at any of these meetings.

19. I also regularly read articles in the scholarly journals that publish material related to the CHI field. I am unaware of any paper that Mr. Galler has ever published on the subject of CHI, in any of the scholarly journals or otherwise.

20. I do not consider Mr. Galler to be an expert in the CHI field in general, or in the design of computer user interfaces in particular. Based on Mr. Galler's academic credentials, with which I am familiar, I believe that other academics and practitioners who work in the CHI field also would not consider Mr. Galler to be an expert in that field. To my knowledge, Mr. Galler has never before held himself out as an expert on CHI issues in general or computer user interface issues in particular.

D. *My Work at the University of Michigan*

21. Throughout my academic career, I have been a cognitive scientist doing research concentrated in the area of computer user interfaces and other cognitive aspects of computers, among other things. In particular, I study CHI issues and their application to the business world. One of my

specialities is cognitive engineering—the use of cognitive psychological principles to design efficient, easy to use computer and other human/machine interfaces for the business environment.

22. Among the courses I teach at the University of Michigan is a high level graduate business school course titled "Human Interface to Information Technology." A significant part of this course is concerned with user interface design, training and measurement in a corporate context.

23. I regularly teach seminars on CHI issues to doctoral candidates. I have acted as a dissertation advisor to more PhD candidates working in computer user interface issues than any other faculty member at the University of Michigan.

24. To my knowledge, Bernard A. Galler has never been asked to serve as a dissertation advisor, or a member of the dissertation committee, for any student working in any aspect of the CHI field.

25. I am regularly asked to referee papers on various CHI issues, particularly as related to computer spreadsheet programs, for the following: Communications of the ACM, ACM Transactions on Office Information Systems, Cognitive Science, Journal of Applied Psychology, Cognitive Psychology, Perception and Psychophysics, Journal of Verbal Learning and Verbal Behavior, Management Information Systems Quarterly. I also regularly review grant proposals for the National Science Foundation, the National Institute of Mental Health, and the National Institute of Education.

E. *Familiarity with Spreadsheet Products*

26. I am very familiar with Lotus 1-2-3, with Borland's Quattro and Quattro Pro spreadsheets, and with several other spreadsheet products. Indeed, Quattro Pro was selected as the recommended spreadsheet for use by students at the University of Michigan.

27. I have spent a significant portion of my academic career studying and evaluating the user interfaces of spreadsheet programs generally, and of Lotus 1-2-3 in particular, using well recognized principles of psychology and business analysis.

F. *The Command Hierarchy of Lotus 1-2-3 Is a System*

28. The command hierarchy of Lotus 1-2-3—the command words and their order—is a system for performing numerical calculations on a spreadsheet. I, along with numerous PhD students under my direction, have for many years applied the principles of cognitive science and human factors analysis to study the efficiency and quality of design of this system.

29. We study the efficiency of the system various ways—for example, we measure the time it takes users to perform tasks with an electronic equivalent of a stopwatch, measuring the time between each keystroke in thousandths of seconds, noting the exact sequence of keystrokes. From this we infer the mental and cognitive processing a user is engaging in while using the system, allowing us to determine at what points the user is examining the menus and at what points the user is performing sequences of keystrokes from memory. Based on this analysis, we determine whether the commands are ordered in a logical, efficient and sensible arrangement. I and several others have published papers devoted solely to the results of this analysis of the Lotus 1-2-3 system.

G. *The Role the Words Play in the Lotus 1-2-3 System*

30. From a cognitive science viewpoint, the step-by-step process that a user goes through in exercising the Lotus 1-2-3 system to accomplish a task is as follows:

(a) First, the user forms a goal that he or she wishes to accomplish using the system (such as insert a new row of cells).

(b) Second, the user engages in a cognitive analysis to determine, either from the choices of functions available in the interface or the system's documentation, the proper command steps that will accomplish the user's goals in this system.

(c) Third, the user presses the necessary keys to activate the functions identified by the words on the screen and evaluates the result.

31. The command words are a central and inseparable part of the Lotus 1-2-3 command system. These words identify the functional choices available to the user, and the system cannot be used without them. The words are required in order for the user to be able to perform the cognitive analysis necessary to determine the command steps that will accomplish the user's goal.

32. The command words do not explain to the user how to use the system—they merely identify the functions that are available in the system. To the extent the user needs explanation and education in using the Lotus 1-2-3 system of commands and in making the correct choice of functions to accomplish a task, that explanation is provided by the documentation for the program, by the help screens, by the tutorials Lotus supplies with the program, and to some extent by the long prompts in the second menu line.

33. Sophisticated users of Lotus 1-2-3 often over time become so familiar with the sequences of command steps necessary to accomplish various tasks that they need no longer examine the words to select the proper steps. Instead, such sophisticated users come to treat the command steps as purely a sequence of keystrokes. These users are able to go directly from the desired goal to typing the keystrokes necessary to accomplish that goal, in much the same way that an experienced typist would automatically type the letters of a word on a typewriter keyboard.

34. Thus, sophisticated users of Lotus 1-2-3 drop the mental process (the cognitive analysis step) in using the command system. For these users, the necessary sequence of commands (and the correlative keystrokes) are merely an automatic stimulus/response sequence that has been learned from intimate familiarity and repeated use of the words of the Lotus 1-2-3 command system. For example, if the sophisticated user desires to insert a new row of cells into the spreadsheet, he or she merely types /WIR on the keyboard, without going through the cognitive analysis of pressing the slash key to bring the menus onto the screen, then selecting the WORKSHEET command, then the INSERT command, followed by the ROW command. Based on many measurements, for these types of users, each successive keystroke requires on average only one third of a second to execute.

H. *The Importance of The Specific Words And Organization*

35. I have read Bernard A. Galler's assertion in ¶ 88 of his Declaration originally submitted in the case of *Lotus Development Corporation v. Paperback Software Int'l*, Civil Action No. 87-0076-K (the "First Galler Declaration"), that possible variations to the command words of a user interface "are limited only by the resourcefulness of developers and . . . by the constraints of language." I believe this assertion is incorrect. To be usable, each of the words in the command hierarchy of a spreadsheet product must convey to the user the function it identifies. With respect to each command there are a very narrow range of words that can perform this function—for example "copy," "replicate," or "duplicate" might serve for the same function. Moreover, shorter, more common words are preferred to enhance recognition and conserve space on the screen.

36. The particular word originally chosen for each function in Lotus 1-2-3 was of no particular importance to users so long as it conveyed the particular function. Lotus 1-2-3 would have been just as successful if it used "replicate" instead

of "copy," or "duplicate" instead of "copy." Of course, the choice of each word had to be consistent with the design rules of the product—two words starting with the same first letter could not be present on the same level of the hierarchy.

37. I have read the Affidavit of Mitchell D. Kapor filed in the case of *Lotus Development Corporation v. Paperback Software Int'l*, Civil Action No. 87-0076-K, in which Kapor states that the commands in the Lotus 1-2-3 hierarchy were organized according to functional considerations such as grouping by functional similarity. For the most part, my own research and observation bears that out, although research has shown that there are somewhat more functional and usable organizations of the commands than the one Kapor adopted.

38. I believe that the particular hierarchical organization of commands adopted by Kapor did not contribute to the success of Lotus 1-2-3. Any functional organization that was as logical and orderly as the one Kapor adopted would have worked just as well, because other aspects of the user interface made the product sufficiently usable to overcome any flaws in the design of the command hierarchy. That the hierarchy itself was not responsible for Lotus 1-2-3's success is demonstrated by the fact that the product was an enormous success, notwithstanding the fact that the hierarchy contained some well documented flaws from a cognitive science point of view.

I. *The Role and Importance of Macros*

39. Although the specific words in the command hierarchy and their specific organization had no particular significance at the time of the development and the initial release of Lotus 1-2-3, the words and their order acquired great commercial significance later because users embedded that system comprised of those words and their order into their own spreadsheet files in the form of macros.

40. I use the term "macro" as defined on page 166 of the Reference Manual for Lotus 1-2-3 Release 2: "A macro is a set of instructions made up of a sequence of keystrokes and commands that you type into a worksheet as cell entries. After you have entered and named the macro, you can invoke it. Whenever you type the macro name, 1-2-3 reads the instructions and performs the specified tasks." Many frequently used macros written by users consist primarily of stored keystroke sequences representing sequences of commands in the menu command hierarchy. However, macros may also contain commands that do not have equivalents in the menu command hierarchy, such as the "/X" commands, which provide certain advanced macro capabilities like branching and customization of menus.

41. People write and use macros for a number of reasons. As stated on page 166 of the Reference Manual for Lotus 1-2-3 Release 2, "Macros are most useful for:

- Automating frequently used 1-2-3 commands
- Typing the same label many times in a worksheet
- Peforming a repetitive procedure that requires a series of sequential commands
- Developing a customized worksheet for someone who is not familiar with 1-2-3"

42. Macros can be *very* long and complicated. For example, the following macro allows someone to extract data from another file (such as a corporate database) and re-format it for use in a Lotus 1-2-3 spreadsheet, which is a common applicaton of macros for business:

```
\i      /fit{?}~
        /dprfcfe{?}~
        i.{end}{down}~
        o{right}~g
```


43. The first line of this macro opens the file in which the desired data is contained, using the name of the file that the user types in response to a prompt. The second line specifies a "recipe" for how the program should break up the incoming data stream into individual items that will be placed into cells of the spreadsheet. The third line instructs the program where to store the incoming data stream that is to be broken up into individual units. The fourth line instructs the program where to place the data after it has been broken up into individual units and then tells the program to perform the breaking up process.

44. Macros are particularly important in corporate environments, as they enable all members of a relevant group to enter, organize, calculate, and present data in the same way without error or deviation. Frequently in a business environment, spreadsheets containing macros written by one user are distributed to other users, who must be able to execute the macros contained in the spreadsheets given to them. For example, the macro described above might be distributed to another person to enable that person to fetch data from another source (such as a departmental database) and place it into that person's spreadsheet for further analysis.

J. *The Role the Words and Their Order Play in Macros*

45. The words of the Lotus 1-2-3 command hierarchy and their order play a central role in the writing and execution of macros. As this Court stated in its opinion in the case of *Lotus Development Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 at 65: "Because macros may contain many menu choices, the exact hierarchy—or structure, sequence and organization—of the menu system is a fundamental part of the functionality of the macros."

46. When a user writes a macro containing Lotus 1-2-3 menu commands, the user stores those commands in the macro in accordance with the word and letter designation and hierarchical order defined by the Lotus 1-2-3 hierarchical

menu command system. In writing the macro, the user uses the words in the menus to select the functional steps that will accomplish the goal the user desires to accomplish and automate, and then embeds those steps in the macro using the command designations defined by those words and in the order dictated by the hierarchical structure of the menus containing the words.

47. The user may also use the words in the menus to debug the macro in the event that it does not run correctly or produces erroneous results. To debug the macro, the user often must step through the macro, one step at a time, watching the menus that come up on the screen to understand what the sequence of commands in the macro is doing and what input may be required by the user when the macro is running, in order to determine whether a step is missing or other error exists in the macro, or whether the steps are not in the proper order as defined by the menu hierarchy.

48. The need to debug macros arises frequently. Macros are often written for a spreadsheet configured in a very specific way. With subsequent use, the configuration of the spreadsheet often gets changed. Even small changes to the spreadsheet, such as the addition of a row or column (a new item in a budget or a new period of time, for example) can make the macro fail to accomplish the desired result. The user must then examine the macro in detail, step by step, to figure out how to modify it to make it work correctly in the new situation, and the menus are needed for this process.

49. Once written, the macro represents a set of steps defined by the user that must be performed in a particular designated way—that is, according to the word and letter designation and hierarchical order of the Lotus 1-2-3 menu command system. Because the user has stored menu commands in the order dictated by the Lotus 1-2-3 menu command system, in order to run the macro to accomplish the user's intended result, the functional steps stored in the macro

must be executed in the particular way defined by that system.

50. Thus, Galler is simply incorrect when he states (at ¶ 129 of the First Galler Decl.),

> There is no technical constraint compelling even a spreadsheet program providing the same functionality as 1-2-3 to do so by using the same set of commands . . .

K. *Dissimilarity of 1-2-3 and Borland Interfaces*

51. From a cognitive science point of view, the user interface of Quattro Pro differs vastly from that of Lotus 1-2-3 in virtually every respect. By "user interface," I mean all aspects by which the user and a computer program interact with one another visually, in terms of physical interaction, and with respect to the approach made available to the user for deciding which functions to invoke to accomplish a goal. The user interface of Lotus 1-2-3 includes the menu hierarchy (the command words and their order), the type of command invocation system (such as pop-up, pull-down, and moving cursor), the long prompts, the help screens, the input mechanisms (such as keyboard and mouse), and other visual aspects of the screen displays produced by the program (such as highlighting of menu items and scrolling down to parts of the spreadsheet not currently visible on the screen).

52. The user interface of Lotus 1-2-3 is very dissimilar—visually, in its physical interaction with users, and in the approach it makes available to the user for solving problems—from the user interface of Quattro Pro in both its native mode and its 1-2-3 compatible mode.

L. *Visual Differences*

53. Visual differences between Lotus 1-2-3 and Quattro Pro are readily apparent from the screen shots described in and submitted with the Declaration of Laurie Flesher in this case. I have examined these screen shots extensively, and I find

them to accurately represent the screens containing the various commands depicted therein. I have also examined the Lotus 1-2-3, Quattro Pro, and Quattro products themselves in forming my opinions.

54. The command invocation system is presented visually to the user in very different ways in the two products because the products are based on entirely different control mechanisms. In Lotus 1-2-3, the command invocation system or control mechanism appears as a set of full word commands displayed in a two-line moving cursor format, in which the first line presents "active" choices of functions that the user may currently select, and the second line often presents "non-active" choices that will become available if the user selects the highlighted command on the first line.

55. Quattro Pro uses an entirely different control mechanism. In Quattro Pro, menu commands are presented inside pull down windows. Commands are presented not only as words but also as graphical images—icons, buttons, and scroll bars. These graphical images, which are entirely absent in Lotus 1-2-3, appear as part of the visual display throughout operation of the program. This same mechanism is used in all product "modes"—Quattro Pro native, Quattro, user-defined, and, of course, 1-2-3 compatible modes.

M. *Differences in Physical Interaction*

56. Quattro Pro's user interface is also radically different in its physical interaction with the user. The command invocation system of Lotus 1-2-3 is designed so that it can be exercised only through use of the keyboard. Commands are selected either by using the cursor keys to highlight the desired command, then pressing enter, or by typing the first letter of the command on the keyboard.

57. In contrast, Quattro Pro is intended to be used with a mouse, in addition to the keyboard, so different physical actions are done by the user, in conjunction with the different visual methods of display. The user may invoke menu com-

mands in one of two ways with the mouse: (a) by clicking and holding the mouse button on a command word to pull down the menu of subcommands under that command, then "dragging" (sliding) the mouse cursor down the list to the desired command and releasing the button, and (b) by clicking and releasing on the menu to pull its window down, then clicking and releasing on the desired command within the menu. Other functions may be invoked by clicking the mouse on icons, buttons and scroll bars.

N. *Differences in Problem Solving Approach Made Available to the User*

58. One of the most significant differences between the user interfaces of Lotus 1-2-3 and Quattro Pro from a cognitive science point of view is the difference in approach each product makes available to the user for solving problems. I am informed that Lotus has attached great significance to the fact that the second line of the menu invocation system presents "non-active" commands—in other words, potential choices of next steps and not active functions.

59. From a cognitive science point of view, I appreciate the significance of this design. The Lotus system is designed in significant part to inform users of what *future* choices will be available to the user for taking the next step after the user selects one of the presently available choices. This information is conveyed through the long prompts, which often list the subcommands that will become available at the next level of the menu hierarchy if the currently highlighted command term is selected at the present level. Thus, at appropriate points, the Lotus system attempts to aid the user in selecting the proper command sequence by showing the user where that sequence is headed next at a given point in the sequence.

60. Quattro Pro presents a very different approach to the user for selecting the proper command sequence. In contrast to Lotus, Quattro Pro almost never displays the next steps that will become available in the sequence after the current step.

Instead, Quattro Pro employs cascading menus that remain on the screen, displaying all *prior* steps to let the user observe the command sequence that led him to a particular point.

61. The Quattro Pro user judges whether he has taken the correct command sequence by examining where he has been, not where he is headed. This difference in approach made available to the user manifests itself in the long prompts. In contrast to Lotus 1-2-3, the long prompts of Quattro Pro almost never present next steps to the user. Rather, they explain the result that will be accomplished if the presently selected command is invoked. It is therefore not surprising that Quattro Pro's long prompts have different words than those of Lotus'—they fulfill an entirely different role.

62. Lotus 1-2-3 and Quattro Pro differ in the approaches they make available to the user for solving problems in one other significant respect. Quattro Pro displays "state" information in the menus, which shows the current state or setting of various command choices, such as margin width, to assist the user in determining whether the settings are correct or might need to be changed. "State" information is not automatically displayed as part of the Lotus 1-2-3 command hierarchy, and in order to discover state, the user must invoke the command that affects that state or, in some instances, invoke the "Status" command.

O. *Differences in Command Hierarchy*

63. I have read Bernard A. Galler's assertions in ¶¶ 29 and 33 of his Declaration submitted in this case that the "menu trees" of Lotus 1-2-3 and Quattro Pro in 1-2-3 compatible mode exhibit "an extremely close parallelism" and are "substantially similar"—indeed, even "virtually identical." I believe that these assertions are incorrect, as a simple examination of the menu trees of the two respective interfaces reveals.

64. I use "menu tree" as defined by Mr. Galler in ¶ 75 of the First Galler Declaration—namely, a visualization of the hierarchical menu structure as a "tree," with dependent commands "branching" from commands at prior levels.

65. I have examined the printouts of the menu trees of Lotus 1-2-3's interface, Quattro Pro's 1-2-3 compatible interface, and Quattro's 1-2-3 compatible interface, described in and submitted with the Declaration of Laurie Flesher in this case.

66. Because of the vast size of the menu trees of Quattro and Quattro Pro, it was not practical to depict them in schematic form. Therefore, the menu trees have been printed in columns of a spreadsheet in which columns to the right of another column contain commands "lower" in the tree than the command in the column to the left. I find that these printouts faithfully set forth and compare the menu trees of the respective products they portray, and accurately illustrate the vast differences in the various command structures.

67. The spreadsheet printouts quite readily reveal the vast dissimilarities in the menu trees of the products. The Lotus 1-2-3 menu tree contains 469 commands. The menu tree of Quattro Pro's 1-2-3 compatible interface contains 5215 commands, a ratio of 11 to 1 more commands. Although each of the 469 commands of Lotus 1-2-3 may be found somewhere within the menu tree of Quattro Pro's 1-2-3 compatible interface, such commands are interpersed at all levels throughout the 5215 commands of the menu tree of the 1-2-3 compatible interface.

68. There is not a single level or layer of the hierarchy in which the Quattro Pro 1-2-3 compatible command hierarchy does not contain menus that are different from those of the Lotus 1-2-3 hierarchy, including the topmost level. Significant differences in the menu trees of the two interfaces exist at all levels or layers.

69. Because of these differences in the command hierarchies and the use of cascading menus, there is no screen comparison of the interface of Lotus 1-2-3 and Quattro Pro's 1-2-3 compatible interface that would reveal identity in visual presentation and in command names in any screen at any point in the hierarchy.

70. The huge number of new commands in the Quattro Pro menu trees are not merely extra choices that have been tacked on to the Lotus 1-2-3 menu tree. Rather, these new functions in Quattro Pro interact with the other elements of the user interface so as to create an entirely different visual perception and physical response by the user, and a different way in which the user interacts with the spreadsheet to get work done.

71. For example, the "View" menu included in the topmost level of the Quattro Pro 1-2-3 compatible menu hierarchy allows the user to change the basic model of presentation of the program—from a full screen single spreadsheet (which is all that is available in Lotus 1-2-3) to a presentation in which multiple spreadsheets can be stacked one on top of another like physical spreadsheets laid out on a desktop, or "tiled" together in a tiling pattern, with edges of adjacent spreadsheets touching one another. The commands within the View menu allow the user to change the appearance and physical interaction with the elements on the screen in the following ways:

- To choose between a "stacked" or "tiled" presentation of spreadsheets
- To pick which of multiple open spreadsheets is to be the "active" one (the one that the user is currently manipulating)
- To "zoom" the currently active spreadsheet to full screen size
- To change the size or position of the active spreadsheet on the screen

72. Moreover, the WINDOW subcommands under the WORKSHEET menu allow the user to change the appearance and physical interaction with the elements on the screen in the following ways, among others:

- To remove the column and row borders from the screen
- To condense the spreadsheet display into a "map view" consisting of columns that are one-character wide and that contain, in place of data, codes that indicate the type of data in each cell.

73. Similarly, the DISPLAY MODE command of the Quattro Pro 1-2-3 compatible interface also enables the user to change many fundamental aspects of the visual presentation of the interface. The user can elect to have the information displayed on the screen in either "character" mode (in which only standard textual letters and characters can be displayed in an arrangement of 25 rows by 80 columns) or in "graphics" mode. In graphics mode, Quattro Pro makes the following display changes:

- If the user has a mouse, the cell pointer will be displayed as an arrowhead rather than as a block.
- If a graph is inserted in the spreadsheet, the user can see the actual graph on the screen when function key 10 is pressed (in character mode, inserted graphs are indicated only by highlighting to show where the graph will appear when printed)
- Some menus will be displayed as "galleries"—depiction of options (such as graph types) as graphical representations rather than words
- The elements of the display appear in finer resolution, and the mouse cursor movements are much smoother on the screen.

74. In sum, the new functionality available through Quattro Pro's 1-2-3 compatible interface not only produces a dif-

ferent visual display, it greatly alters the way the user can interact with the program.

P. *Differences in Quattro*

75. Much of the analysis and conclusions I have described with respect the user interfaces of Quattro Pro also apply to Quattro.

76. Quattro used yet a different command invocation mechanism—"pop up" menus. But, like Quattro Pro, Quattro does not show the user where he is "headed" by displaying the next steps in the sequence of commands that will become available if the currently selected command is invoked. Instead, it presents purely "present" command choice information, displaying at any given time only a single pop up menu showing the presently available "active" commands.

77. Like Quattro Pro, the long prompts of Quattro are almost always explanatory of the current active function choices. And, again as in the case of Quattro Pro, Quattro provided other explanatory aspects in its interface, such as help screens and current state information—all of which are either different from Lotus 1-2-3 (as in the case of help screens) or have no correlative in Lotus 1-2-3 (as in the case of state information).

78. With respect to the menu hierarchy, the menu tree of Quattro's 1-2-3 compatible interface contains 3370 commands, compared to 469 commands in the Lotus 1-2-3 menu tree—a ratio of 7 to 1 more commands. Although each of the 469 commands of Lotus 1-2-3 may be found somewhere within the menu tree of Quattro's 1-2-3 compatible interface, such commands are interpersed at all levels throughout the 3370 commands of the menu tree of the 1-2-3 compatible interface. There is not a single level of the hierarchy in which the Quattro 1-2-3 compatible command hierarchy is the same as the Lotus 1-2-3 hierarchy, including the topmost level. Significant differences in the menu trees of the two interfaces exist at all levels.

Q. *The Role of the Native Interface and the 1-2-3 Compatible Interface*

79. Based on human factors analysis of the Lotus 1-2-3 interface, the Borland 1-2-3 compatible interfaces, and the Borland native interfaces, it is difficult to imagine that users will find the 1-2-3 compatible interfaces easy to use for anything other than writing, debugging or executing macros.

80. The Lotus 1-2-3 user interface has well known design flaws. For example:

• Patterning after the English language, users generally tend to think of goals they wish to accomplish in the spreadsheet in a "verb-object-modifier" sequence. Thus, a user's goal might be to "retrieve (verb) a file (object) named TEXT (modifier)." A menu structure organized to enable a user to invoke commands in this order feels "natural" to the user, for it comports with the way a user tends to think of his or her goal in the mind.

The menu structure of Lotus 1-2-3, however, is not consistently organized in this way. Lotus 1-2-3 often requires the user to invoke commands in a "object-verb-modifier" sequence to accomplish many goals, which feels less "natural" to the user. For example, the commands FILE RETRIEVE NAME, WORKSHEET ERASE and DATA SORT are all commands that are of the "object-verb-modifier" type. Other goals, however, do require a "verb-object-modifier" sequence, thereby mixing up the order in which the user must think about command invocation for various different tasks. For example, the commands ERASE WORKSEET and INSERT ROW are commands that are of the "object-verb-modifier" type. This inconsistency in structure makes the Lotus 1-2-3 command hierarchy more difficult to learn and to master.

• The second line of the two-line moving cursor, which often shows commands that will become available in the next level down of the menu structure if the currently highlighted command is selected, is often confusing to new users. New users frequently think that the commands listed in the second line present additional "active" choices that can be invoked at the current level of the menu structure. Thus, users often type the first letter of a command listed in the second line, which results either in the program issuing a "beep" (to indicate an illegal letter choice), or activation of an unintended command on the first line, causing undesired consequences that the user may then have to repair.

81. When all the enhanced functionality of the Borland product is added into the Lotus 1-2-3 interface at various points in the 1-2-3 compatible interface, the 1-2-3 compatible interface becomes awkward, difficult and inefficient to use as a system for solving numerical problems.

82. Such awkwardness and inefficiency results principally for two reasons. First, addition of new functions into the existing Lotus 1-2-3 command hierarchy often effectively changes the meaning of existing menus, as illustrated by the additions to the Lotus WORKSHEET menu in the 1-2-3 compatible interface. Second, such additions often cause confusion as to where to look for the functionality residing in previously existing menus, as illustrated by Borland's new VIEW menu.

83. Because Lotus 1-2-3 was designed with two-line moving cursor menus, in which commands are arranged horizontally across the screen, the number of commands that can be placed in any given menu is limited by the width of the screen to about ten. As a result of this constraint, the designers of the Lotus 1-2-3 menus were forced to group commands together under a limited number of top level menus. Accordingly, the WORKSHEET top level menu was created to group together a number of miscellaneous commands that, although very diverse in their functionality, affected the entire worksheet.

84. Because much of the new functionality of Borland's Quattro and Quattro Pro products has no correlative in Lotus 1-2-3, such functionality does not really fit well within any of the Lotus 1-2-3 top level menu categories. Nevertheless, because screen width limitations severely limit the addition of new top level menus, the new functionality had for the most part to be placed somewhere within the existing Lotus 1-2-3 top level menus. In particular, much of the new functionality had to be put in the WORKSHEET menu, because that is the only top level Lotus menu akin to a "Miscellaneous" category—all other top level Lotus 1-2-3 menus relate to very specific types of functionality, such as COPY, MOVE, FILE, PRINT and the like.

85. The addition of the new Borland functionaltiy to the WORKSHEET menu changes substantially the meaning of that menu. Previously, it grouped together diverse functions that all related to entering data into, editing and styling the spreadsheet itself. Many of the new Borland functions that were added to this menu, however, are not of this type—such as UNDO and MACRO commands. Thus, the presence of the new Borland functions in the WORKSHEET menu effectively changes its meaning from a category of "WORKSHEET" functions to a category of "OTHER" functions. The user now is unsure of what types of "other" functions that menu contains, which breeds inefficiency and awkwardness in looking for functions.

86. A second illustration of the awkwardness and inefficiency that results from addition of substantial new functionality to the Lotus 1-2-3 command hierarchy is the new VIEW menu. The VIEW menu commands allow the user to see and manipulate multiple spreadsheets on the screen simultaneously. Lotus 1-2-3 has a WORKSHEET WINDOW menu, which contains commands that allow the user to split a single worksheet into two parts and view the parts simultaneously. Because the WORKSHEET WINDOW menu commands all affect only a single worksheet, that menu did not easily acco-

modate the new VIEW commands, which affect multiple worksheets.

87. Accordingly, Borland created a new menu titled "VIEW" at the topmost level of the 1-2-3 compatible command hierarchy. Because this new menu is at the topmost level, it is apt to confuse the user as to where to look for the old functionality. In particular, I would expect that many users will be misled to look for the old WORKSHEET WINDOW functions in the VIEW menu, rather than in their original location in the Lotus 1-2-3 hierarchy.

88. In sum, users will find it awkward to access the new functionality of the Borland products, as well as the original Lotus 1-2-3 functionality, through the 1-2-3 compatible interface because of its poor human factors design.

89. From a human factors point of view, the native interfaces of the Borland products are quite superior to the 1-2-3 compatible interfaces. For example:

• *Conceptual Grouping*. There are two general kinds of activities a user may engage in while using a spreadsheet: (a) entering data and performing calculations on that data, and (b) formatting the data for a pleasing and easy-to-read appearance. The Borland native interfaces group the commands for each of these kinds of activities consistently together each under its own menu (under the EDIT and STYLE menus, respectively). The 1-2-3 compatible interfaces do not, but rather spread the two kinds of activities throughout different menus and place commands relating to both kinds of activities within the same menu. For example, commands for entering data and performing calculations are contained in the WORKSHEET, RANGE, COPY and MOVE menus, which also contain commands for formatting the data. Thus, the user is forced to bounce around through multiple menu subtrees no matter what type of major subtask the user is attempting to accomplish.

• *Additional Visual Cues That Aid Organization*. The native interface menus present additional visual cues to the user that make the organization of the command choices clearer to the user. For example, under the STYLE menu in the native interface, there is a cluster of menu choices that all relate to columns, and these choices are separated from the other choices within the menu by a horizontal line. Similarly, in the FILE menu, items that a user invokes when beginning work on a file (NEW, OPEN and RETRIEVE) are clustered together and separated by a horizontal line from those items that a user invokes when ending work on a file (SAVE, SAVE AS, CLOSE and the like). These additional visual organizational cues are absent entirely from the 1-2-3 compatible interface menus of Quattro. Although present to a limited extent in the 1-2-3 compatible interface menus of Quattro Pro, these visual cues in most cases do nothing more than separate the new commands that Borland has added to that interface that have no equivalent in Lotus 1-2-3.

90. Because, from a human factors viewpoint, the Borland native interfaces are much better designed and easier to use than the 1-2-3 compatible interfaces, I believe that users of the Borland products would naturally be inclined to use the native interface for ordinary spreadsheet work, and not the 1-2-3 compatible interfaces. I would expect that the 1-2-3 compatible interfaces will be used only to write, debug and execute Lotus 1-2-3 macros.

R. *The "Mental Model" of Lotus 1-2-3*

91. I have read ¶ 30 of the First Galler Declaration in which Mr. Galler asserts that in designing a computer program the developer should create a "mental model" for the program, which Mr. Galler defines as a "conception of the program's behavior, which the user is intended to form in operating the program based upon the user's perception of the user interface and externals." Later, in ¶ 106 of his Affidavit, Mr. Galler

asserts that the menu structure of Lotus 1-2-3 presents the mental model of Lotus 1-2-3 to the user.

92. Mr. Galler's definition of "mental model" does not comport with how that term is generally defined in the CHI field. Although I am familiar with the scholarly literature in the CHI field discussing "mental models," and have myself written papers on that subject, I have never heard anyone in the field use that term to include the menu commands of a user interface.

93. With respect to computer user interfaces, a "mental model" is usually defined to mean a metaphor formed in the user's mind about the internal aspects of the system that aids the user in selecting the proper commands in the interface and otherwise making use of the computer program to accomplish a task. The mental model underlying both Lotus 1-2-3 and the Borland products is the spreadsheet metaphor—a collection of cells in spatial relationships that can contain numbers, formulas and cross-references among themselves, as well as, in the case of macros, commands.

94. Under this definition, the menu commands of Lotus 1-2-3 do not comprise or convey a mental model, for the commands themselves do not aid the user in determining which commands should be chosen, or in what sequence.

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury under the laws of the United States this 26 day of September 1991 at Palo Alto, California.

/s/ JUDITH OLSON
Judith Olson

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.

Defendant.

DECLARATION OF VERN L. RABURN

I, VERN L. RABURN, hereby declare under penalty of perjury, as follows:

A. *Identity of Declarant*

1. I am the chairman and chief executive officer of Slate Corporation, a company I co-founded in 1990 to develop software applications for pen-based computer systems.

2. In 1982, I joined Lotus Development Corporation ("Lotus") as executive vice-president and general manager. At Lotus, I had ultimate responsibility for the "launch" and initial marketing of Lotus 1-2-3.

3. Prior to my tenure at Lotus, I served as President of the Consumer Products Division of Microsoft Corporation, located in Bellevue, Washington. In this capacity, I organized

and directed Microsoft's entry into applications software development and retail software distribution.

B. *Other Experience*

4. I hold a B.S. degree in industrial technology from California State University at Long Beach. In 1976, I opened one of the nation's first computer stores and went on to become operations manager for Byte Industries. After leaving Byte Industries, I formed a software division for GRT Corporation, the first company to license, manufacture and market packaged consumer software for personal computers.

5. After leaving Lotus in 1982, I joined Symantec Corporation, as Vice-President of Marketing. Symantec develops, manufactures and markets microcomputer software. I served as president and chief executive officer of Symantec from 1983 to 1987 and as chairman of Symantec's board of directors from 1986 to 1990.

6. From 1987 to 1989, I was also a general partner in the venture capital firm of Cooper, Raburn & Kniffin, which specialized in funding and nurturing very early-stage, high technology start-up companies.

C. *Responsibilities at Lotus*

7. I was personally recruited by Mitch Kapor to join the management of Lotus, then a startup company, in 1982, and I joined the company prior to the launch of 1-2-3 in October of that year. I was given the title of executive vice-president and general manager and given primary responsibility for all aspects of Lotus' business, with the exception of product development and accounting, which continued to be Mr. Kapor's responsibility.

8. As part of my responsibilities at Lotus, I had ultimate responsibility for devising and implementing the initial product "roll out" or "launch" of 1-2-3, the initial marketing plan for the product and the product's initial distribution plan. I

successfully negotiated Lotus' initial exclusive distribution agreement with Softsel Corporation, then the largest software distributor.

9. A vital component of 1-2-3's marketing plan, and, hence, a key component of my job was to create interest in and demand for 1-2-3, by giving product demonstrations and presentations to distributor executives, personal computer software dealers and potential users. During the months of November and December of 1982 and January of 1983 much of my time was consumed with product demonstrations and presentations; I gave approximately 150 such demonstrations prior to the first commercial shipment of 1-2-3 on January 26, 1983.

D. *Perceived Value of the Command Hierarchy*

10. It was my goal in these presentations to identify and demonstrate those features of 1-2-3 that my staff and I believed were most likely to interest potential users in the purchase of the product and which were likely to be perceived by potential users as important advancements beyond features available in the leading competitive product, VisiCalc. In each presentation, it was my practice to ask for questions and comments from those in attendance. As time went on, I continued to modify my presentation to emphasize those features that prior attendees had indicated was of greatest interest to them.

11. The "command hierarchy" of 1-2-3 played no ascertainable role in the commercial success of the product. The "command hierarchy" was not identified and presented by me or my staff as an important feature of the product, nor did dealers, potential users or others in attendance identify the command hierarchy as important to them. By "command hierarchy," I mean the particular hierarchical organization of command names in 1-2-3.

12. The "command hierarchy" portion of the Lotus 1-2-3 user interface is a system—a particular step by step approach—designed by Kapor and Sachs for the solution of numerical problems using a spreadsheet. Indeed, the Release 1.0 users' manual for 1-2-3 unambiguously characterized the hierarchy as a "system":

> 1-2-3's commands are organized into a multi-level menu system.
>
> Release 1.0 Users' Manual at 53. (Attached as Exh. A.)

13. Lotus' 1-2-3 was demonstrated by me and others in the company as a solution to some of the principal functional problems of VisiCalc. In my view, it succeeded commercially for this reason. The key features of 1-2-3 that I emphasized and attendees asked about in demonstrations included (roughly in order of importance):

(a) variable column widths in the spreadsheet;

(b) speed of the product for moving around the spreadsheet and recalculation;

(c) the user's ability to highlight within a specified "range" of the spreadsheet;

(d) full words in the screen presentation with progressive disclosure of additional functions;

(e) integrated graphics capabilities; and

(f) data manipulation capabilities.

14. The specific words in the command hierarchy of 1-2-3 were of no particular consequence to me, my staff or potential users when I demonstrated the product. Any word would have sufficed for a command so long as the word conveyed to the user in a direct and simple way the function of the command. This was true with respect to all spreadsheet products (and, for that matter, application programs generally) not just Lotus 1-2-3. For example, VisiCalc used the command "R" for

"replicate" in its command hierarchy because commands were listed alphabetically and the "C" command was previously used for "column." Yet the choice of "replicate" as opposed to "copy" had no commercial effect or significance because both words convey the functionality of the command.

15. Similarly, the precise order of the commands in the hierarchy was of no particular importance to me, my staff or attendees at the demonstrations. The fact that 1-2-3 commands were listed in a functional ordering as opposed to alphabetically was an important departure from VisiCalc, but the precise order of the commands could certainly have varied without consequences. Potential users would have readily accepted any easily useable, functional grouping of commands because the advancements in other aspects of 1-2-3 so significantly overshadowed VisiCalc.

16. The initial advertising plan for 1-2-3, as implemented by me, touted "ease of learning" as an important feature of the product. The basic advertising and marketing slogan for 1-2-3 was "easier, faster, better." As the particular ads we published explain, "ease of learning" was a reference to context-sensitive on-screen "help" instructions and explanations, "long prompts," the ability of the user to highlight and work within ranges, as well as the fact that commands were set forth in full words rather than just first letters. The "command hierarchy"—the particular command names and the precise order of those commands—was never an ease-of-learning feature that I recall, nor was it considered an "ease-of-learning" feature by me.

E. *Macros in 1-2-3*

17. "Macros" or "keyboard macros" defined in the Lotus 1-2-3 Release 1.0 user's manual as a "stored sequence of keystrokes," were not part of the product development specification for 1-2-3 as envisioned and prepared by Mitch Kapor. Nor was the macro capability requested by potential customers. Rather, Mitch Kapor added a macro capability as

a free feature for marketing reasons. He wanted 1-2-3 to include the same functionality as a popular program known as "Prokey," which allowed users to record sequences of keystrokes.

18. The macro capability in 1-2-3 was originally conceived merely as an automated keystroke system. The 1-2-3 user's manual for Release 1.0 even refers to keyboard macros as the "Typing Alternative." 1-2-3 User's Manual Release 1.0 at 107 (Attached as Exh. B). The manual instructs users to use macros to perform "procedures" consisting of several commands "in a particular order":

> There are many situations in which you'll find the Typing Alternative useful:
>
> * * *
>
> • You need to perform a certain procedure, involving several commands and/or entries in a particular order.
>
> *Id.* at 107.

F. *Relative Importance of Code and UI*

19. I have read ¶¶ 28-52 of the Declaration of Bernard A. Galler originally filed in the case of *Lotus Development Corporation v. Paperback Software International*, Civil Action No. 87-0076-K, and I am informed as to the significance he attached to the user interface of a software product, as opposed to its code. Based on my extensive experience in the marketing and sale of software products, I believe it is fundamentally incorrect to suggest that the code of a software product is unimportant or always less important than the user interface. Of course, for some products the user interface of the product may be extremely important to its commercial success—entertainment software is a good example of such a product. In other cases, compilers for example, the user interface may be minimal, unimportant or even nonexistent. And,

in any case, a product that cannot supply necessary functional features will always fail, regardless of the interface.

20. It is my experience that in more advanced and sophisticated software development, when the user interface is perceived to be a key to the product's success, the development of the user interface will precede or move in parallel with the development of the code.

21. I believe that the success of 1-2-3 was due in significant part to the computational and screen display code written by Jonathan Sachs that made 1-2-3 a faster and more robust product than VisiCalc.

G. *Legal Protection for Command Hierarchies*

22. It is my understanding that, generally speaking, patents can be secured in appropriate circumstances to preclude a competitor from making products that users can operate in a manner similar to the product on which patent protection is obtained. I am also informed that copyright protection is relatively easy to secure, while patent protection is, generally speaking, far more difficult to obtain.

23. I have been involved in the personal computer industry for more than 15 years and I have held positions in the top management of a number of the industry's leading software publishers. Based on my experience, I believe that extending copyright protection to the command words and their order of a program like Lotus 1-2-3 will not promote investment, innovation, creativity or competition in the software industry generally or with respect to spreadsheet products specifically. To the contrary, extending easily obtainable monopoly-like protection will do nothing more than enable the first company that establishes a widely successful product having a particular set of ordered command words to lock users in to that company as the sole source of products.

24. Because a substantial base of users may be tied to the use of a particular system of ordered words, in order to run

existing macros or to make use of learned keystrokes to invoke functionality, permitting a company to monopolize a particular order of commands will enable it to effectively shut out competition from other companies that might provide superior products to that same base of users.

I declare that the foregoing is true and correct, and that I have personal knowledge of the matters set forth above. If called as a witness, I could and would competently so testify.

Executed under penalty of perjury under the laws of the United States this 22 day of September, 1991 at Santa Ana, California.

/s/ VERN L. RABURN
Vern L. Raburn

[EXHIBIT A]

Using 1-2-3 Commands

During every 1-2-3 session, you issue commands. For instance, the way to end a 1-2-3 session is to issue the /Quit command. Other 1-2-3 commands perform such functions as:

- Copying, moving, and deleting data from the worksheet.
- Transferring data between the worksheet and disk storage.
- Printing reports.
- Drawing graphs.
- Handling databases.

Complete details concerning the individual 1-2-3 commands appear in the "Using Commands" section of this manual. In this section, we discuss the general aspects of issuing commands.

1-2-3 commands are easy to use because they're interactive. A command may have one or more steps. At each step, 1-2-3 tells you what to enter next: a number, a range, a filename, etc. Messages from 1-2-3, called prompts, appear on the second line of the control panel. Here are some typical ones:

Enter range to copy FROM:

Enter name of file to retrieve:

Enter Sort order (A or D):

After you've entered all the specifications, 1-2-3 executes the command and (usually) returns to Ready Mode. Let's take a closer look at each of the steps involved in issuing a 1-2-3 command.

Selecting a Command

Each 1-2-3 command has a name, consisting of one or more command keywords, such as:

/Copy

/Range Erase

/Worksheet Column-Width Set

/Quit

/Graph Options Titles

1-2-3's commands are organized into a multilevel menu system. You build up a command name by selecting one keyword at a time from menus that 1-2-3 displays in the control panel. The only thing that 1-2-3 doesn't automatically tell you is how to start:

[EXHIBIT B]

Keyboard Macros—
The Typing Alternative

1-2-3 has a facility that allows you to store sequences of keystrokes for future use—during the same session or during a later one. The keystrokes might invoke one or more formatting commands, type entries, create a graph, etc.

In computer circles, a stored sequence of keystrokes is often called a keyboard macro. We'll also use the shortened form, macro.

You "attach" each macro to one of the letter keys: A,B,C, . . . Z. When you hold down the [Alt] key and type the letter, 1-2-3 automatically types all the stored keystrokes in the sequence, just as if you had typed them again yourself.

Since this facility allows you to save keystrokes, we call it the "Typing Alternative." But we won't mind if you do as the programmers do—call it "the macro facility."

There are many situations in which you'll find the Typing Alternative useful:

- You need to type the same label many times in a worksheet. For instance, "Revenues", "1st Quarter", or your company name.
- You need to perform a certain procedure, involving several commands and/or entries in a particular order. For instance, a macro might produce a printed report consisting of several sections of the worksheet, each with its own margins, character size, line spacing, etc. [A sequence of commands is a "procedure"]*
- In a worksheet that you're developing for use by "non-technical" personnel, you wish to use some of 1-2-3's more

---

* Bracketed portion is handwritten in original.

sophisticated functions. But you can't afford the time to train the personnel in the program's advanced features.

Storing the Keystrokes

We suggest that you go through a "dry run" before using the Typing Alternative: Type the entries and/or commands, noting exactly which keys you press. Don't forget to note—[Enter] and [Space Bar]! When you're satisfied that you know precisely which keystrokes are to be saved, find some out-of-the-way cells, in an empty part of the worksheet. Then:

1. Enter the macro as one or more labels (e.g. preceded by a label-prefix).

2. Assign a special macro range name to the cell in which the keystroke sequence begins.

We'll examine both these steps in detail, then proceed to a discussion of using (invoking) macros.

[107]

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 87-0076-K
Filed in Open Court

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

PAPERBACK SOFTWARE INTERNATIONAL
and STEPHENSON SOFTWARE, LTD.,

Defendants.

Civil Action No. 87-0074-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

MOSAIC SOFTWARE, INC.,

Defendant.

AFFIDAVIT OF JONATHAN M. SACHS

JONATHAN M. SACHS, being duly sworn, deposes and says:

1. My name is Jonathan M. Sachs and I reside in Cambridge, Massachusetts.

2. I make this affidavit to provide direct testimony in the Phase I trial in the above-captioned actions.

3. Mitchell Kapor ("Kapor") and I are the authors and creators of the computer program known as Lotus 1-2-3 (also "1-2-3").

4. The version of Lotus 1-2-3 first sold to the public on January 26, 1983—called Release 1.0—was the result of a process of collaboration between Kapor and myself lasting approximately eighteen months.

5. In developing Lotus 1-2-3, I was primarily responsible for writing the program's source code and Kapor was primarily responsible for its design, although I also contributed to the design of the program.

6. During the period of 1-2-3's creation, Kapor and I discussed hundreds of ideas concerning possible features to include in our product and ways to express the program's functionality through its user interface. Typically, Kapor suggested a feature; we discussed it; and I attempted to implement it in the program's source code. We then reviewed the results and, more often than not, changed it again. We repeated this process countless times during this period.

7. Kapor had principal responsibility for creating the user interface of 1-2-3 and, in particular, the content and organization of its menus. My principal role concerning these aspects of the product was to ensure that the program's menus and user interface were successfully implemented in its source code.

8. During the last several months before 1-2-3 was finished I, Kapor and others at Lotus participated in a long series of

meetings, the purpose of which was to finalize and rationalize the contents and organization of the menus. Kapor ultimately decided what words would be used and where on the menu tree they would be located. I then modified the source code to reflect his changes. Again, in this period Kapor was more concerned about the user interface and I was more concerned about the program's internal workings.

9. After months of brainstorming, experimentation and implementation, Lotus 1-2-3 was commercially released. The finished product was the result of the original creative efforts of Kapor and myself (and, to a lesser extent, the other members of our development team).

Sworn to under the penalties of perjury this 27th day of November, 1989.

/s/ JONATHAN M. SACHS
Jonathan M. Sachs

BEST AVAILABLE COPY

(8)

No. 94-2003

Supreme Court, U.S.
FILED
NOV 9 1995
CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

JOINT APPENDIX
VOLUME II OF II
(Pages J.A. 540 to J.A. 985)

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

GARY L. REBACK
Counsel of Record
MICHAEL BARCLAY
SUSAN A. CREIGHTON
COLLEEN BAL
WILSON, SONSINI, GOODRICH & ROSATI
Professional Corporation
650 Page Mill Road
Palo Alto, California 94304
(415) 493-9300

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY & GELHAAR
One Post Office Square
Boston, Massachusetts 02109-2105
(617) 542-7400

Counsel for Respondent

(Petitioner's counsel continued on inside front cover)

PETITION FOR CERTIORARI FILED JUNE 7, 1995
CERTIORARI GRANTED SEPTEMBER 27, 1995

459 MP

NEAL D. GOLDMAN
LOTUS DEVELOPMENT
CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-1127

DONALD J. ROSENBERG
International Business Machines
Corporation
44 South Broadway
White Plains, New York 10601
(914) 288-4010

Counsel for Petitioner

TABLE OF CONTENTS

PAGE

PLEADINGS

Docket Entries ..... J.A. 1

Complaint, filed July 2, 1990 (Dkt. No. 1) ..... J.A. 13

Answer and Demand for Jury Trial, filed July 31, 1990 (Dkt. No. 4) ..... J.A. 21

Brief Amicus Curiae On Behalf Of Register Of Copyrights, filed September 30, 1991 (Dkt. No. 85) ..... J.A. 27

Supplemental Complaint, filed January 19, 1993 (Dkt. No. 305) ..... J.A. 32

Answer to Supplemental Complaint, filed January 21, 1993 (Dkt. No. 307) ..... J.A. 36

Stipulation and Order Regulating Trial, filed February 3, 1993 (Dkt. No. 330) ..... J.A. 40

Borland's Motion To Amend Original Answer, filed February 26, 1993, with Proposed Amended Answer annexed thereto (Dkt. No. 341) ..... J.A. 56

Notice of filing of Trial Record, filed May 5, 1993 (Dkt. No. 359) ..... J.A. 64

Notice of filing of "Software Library," filed May 5, 1993 (Dkt. No. 360) ..... J.A. 68

PAGE

HEARING & TRIAL TRANSCRIPTS

Lotus v. Paperback Trial Transcript, dated March 22, 1990, Vol. XII (Submitted with Dkt. No. 200) ...J.A. 71

Conference, dated June 18, 1991 (Dkt. No. 66) .....J.A. 74

Memorandum and Order, dated March 20, 1992 (Dkt. No. 164).....J.A. 77

Hearing on Cross-Motions for Summary Judgment, dated May 19, 1992 (Dkt. No. 194) .....J.A. 80

Procedural Order with Exhibits thereto, dated September 30, 1992 (Dkt. No. 232) .....J.A. 86

Non-Jury Trial Proceedings Day 3, dated February 3, 1993 (Dkt. No. 333).....J.A. 106

Non-Jury Trial Proceedings Day 4, dated March 31, 1993 (Dkt. No. 358).....J.A. 157

Non-Jury Trial Proceedings Day 5, dated April 1, 1993 (Dkt. No. 404).....J.A. 279

Non-Jury Trial Proceedings Day 6, dated April 2, 1993 (Dkt. No. 405).....J.A. 301

AFFIDAVITS & DECLARATIONS
(Submitted with Dkt. No. 359 unless noted otherwise)

Declaration of Timothy A. Boyd, dated September 26, 1991 (Submitted with Dkt. No. 89) .....J.A. 307

PAGE

Declaration of Eugene K. Buechele, dated September 27, 1991 .....J.A. 310

Second Declaration of Eugene K. Buechele, dated August 27, 1992 (Submitted with Dkt. No. 199) J.A. 313

Declaration of Jennifer Dos Remedios, dated November 1, 1991.....J.A. 317

Declaration of James C. Emery, dated November 1, 1991.....J.A. 320

Declaration of Bernard A. Galler in *Lotus v. Paperback*, dated November 27, 1989.....J.A. 357

Second Declaration of Bernard A. Galler, dated November 1, 1991.....J.A. 382

Declaration of Marvin Goldschmitt, dated September 24, 1991 (Submitted with Dkt. No. 89) .....J.A. 399

Declaration of David E. Kieras, dated September 26, 1991.....J.A. 407

Declaration of Robert H. Kohn, dated September 27, 1991.....J.A. 419

Supplemental Declaration of Robert H. Kohn, dated January 25, 1993 .....J.A. 428

Declaration of Thomas M. Lemberg, dated May 6, 1991.....J.A. 435

Declaration of David E. Liddle, dated September 27, 1991.....J.A. 439

PAGE

Declaration of Jim P. Manzi, dated November 1, 1991 ..... J.A. 470

Declaration of Arthur R. Miller, dated January 3, 1985 (Addendum to Borland's Reply Brief before the First Circuit) ..... J.A. 476

Second Declaration of Arthur R. Miller, dated May 20, 1985 (Addendum to Borland's Reply Brief before the First Circuit) ..... J.A. 487

Declaration of Judith S. Olson, dated September 26, 1991 ..... J.A. 498

Declaration of Vern L. Raburn, dated September 22, 1991, with Exhibits A and B annexed thereto (Submitted with Dkt. No. 89) ..... J.A. 525

Affidavit of Jonathan M. Sachs in *Lotus v. Paperback*, dated November 27, 1989 (Submitted with Dkt. No. 36) ..... J.A. 537

DEPOSITIONS
(Submitted with Dkt. No. 359 unless noted otherwise)

Deposition of Adam Bosworth, April 29, 1991 ..... J.A. 540

Deposition of Adam Bosworth, April 30, 1991 ..... J.A. 567

Deposition of Robert H. Dickerson, September 12, 1991 ..... J.A. 592

Deposition of James Emery, November 8, 1991 ..... J.A. 597

PAGE

Deposition of Bernard A. Galler in *Lotus v. Paperback*, January 25, 1990 (Submitted with Dkt. No. 92) ..... J.A. 602

Deposition of Bernard A. Galler, August 28, 1991 ... J.A. 606

Deposition of Bernard A. Galler, August 29, 1991 ... J.A. 624

Deposition of Bernard A. Galler, August 30, 1991 ... J.A. 629

Deposition of Ezra Gottheil, September 22, 1992 .... J.A. 632

Deposition of Frank A. Ingari, August 6, 1991 (Submitted with Dkt. No. 115) ..... J.A. 648

Deposition of Becky D. Jones, September 5, 1991 (Submitted with Dkt. No. 115) ..... J.A. 652

Deposition of Philippe Kahn, May 22, 1991 ..... J.A. 655

Deposition of David E. Kieras, October 7, 1991 ..... J.A. 663

Deposition of W. Frank King, September 12, 1991 .. J.A. 674

Deposition of Robert H. Kohn, September 11, 1991 . J.A. 678

Deposition of Thomas M. Lemberg, July 3, 1991, with Exhibit 13 annexed thereto ..... J.A. 686

Deposition of David E. Liddle, October 10, 1991 .... J.A. 692

Deposition of Joanne H. Lin, September 6, 1991 .... J.A. 706

Deposition of Jim P. Manzi in *Lotus v. Paperback*, December 30, 1987 ..... J.A. 709

PAGE

Deposition of Jim P. Manzi, August 22, 1991 .....J.A. 712

Deposition of Judith S. Olson, September 26, 1991 ..J.A. 717

Deposition of Judith S. Olson, October 15, 1991 ....J.A. 734

Deposition of Jack Oswald, August 29, 1991 .....J.A. 736

Deposition of John Posner in *Lotus v. Paperback*, March 7, 1989.....J.A. 743

Deposition of Jonathan M. Sachs in *Lotus v. Paperback*, November 29, 1988.....J.A. 749

Deposition of Dorothy Schrader, September 13, 1991.....J.A. 751

Deposition of Robert W. Warfield, May 15, 1991 ....J.A. 754

Deposition of Robert W. Warfield, May 16, 1991 ....J.A. 776

Deposition of Robert W. Warfield, November 9, 1992.....J.A. 784

EXHIBITS

Exhibits in Support of Lotus' Motion for Summary Judgment on Liability, filed May 7, 1991 (Declaration of Kerry L. Konrad (Dkt. Nos. 36 and 148))

Exhibit 12 .....J.A. 801

Exhibit 26 .....J.A. 809

Exhibit 30 .....J.A. 811

PAGE

Exhibit C-27 .....J.A. 814

Exhibit to Lotus' Motion for Summary Judgment, filed November 1, 1991 (Declaration of Paul M. O'Connor III (Dkt. No. 108))

Exhibit C .....J.A. 817

Exhibits in Support of Borland's Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment, dated January 30, 1992 (Dkt. No. 142)

Ex. 1: Document entitled "Quattro Pro Honors: Awards and Quotes" .....J.A. 821

Ex. 6: Letter from the Copyright Office to Kerry L. Konrad .....J.A. 827

Ex. 9: Lotus User's Manual .....J.A. 829

Ex. 14: Document entitled "Interoperability Qualitative Study" .....J.A. 831

Ex. 15: Lotus European Patent Application .....J.A. 837

Ex. 16: U.S. Patent No. 4,989,141 .....J.A. 842

Ex. 19: Letter from the Copyright Office .....J.A. 847

Ex. 20: Document from the Copyright Office entitled "Computer Programs and Related Screen Displays." .....J.A. 849

PAGE

Ex. 21: Appellants' Opening Brief in *Ashton-Tate Corp. v. Ross*, No. 89-15683 (9th Cir.).....J.A. 853

Ex. 23: U.S. Patent No. 4,611,306 .....J.A. 856

Ex. 27: Document entitled "Quattro Pro Displacement Study".....J.A. 864

Ex. 36: Text of Jim Manzi's Royal Sonesta Staff Meeting speech .....J.A. 873

Ex. 38: Document entitled "Houdini Analysis of Competitive Products".....J.A. 877

Ex. 39: Memorandum entitled "Review of Excel" ...J.A. 879

Ex. 40: Memorandum entitled "1-2-3 3.0 Positioning Study".....J.A. 881

Ex. 41: Memorandum entitled "Quattro Evaluation"..J.A. 885

Exhibit in Support of Borland International, Inc.'s Motion for Reconsideration, dated August 31, 1992 (Declaration of Andrew Konstantaras (Dkt. No. 200))

Ex. I.....J.A. 887

Documentary Exhibits Submitted at Trial, filed May 5, 1993 (Dkt. No. 359)

Ex. 7: Appendix I from Quattro Pro 3.0 User's Manual.....J.A. 899

Ex. 15: Document bearing Borland production Nos. 503897-503898 .....J.A. 907

PAGE

Ex. 18: Telegram from Philippe Kahn (Borland President) to Henry B. Gutman (counsel for Lotus) .J.A. 909

Ex. 19: Letter dated September 6, 1990 from Philippe Kahn (Borland President) to Dorothy Konrad.....J.A. 912

Ex. 32: Memorandum regarding CR4314 dated July 31, 1992.....J.A. 915

Ex. 33: Internal Bug Report .....J.A. 916

Ex. 36: Dybvig, Phillip H., *The Lotus Tutorial*.....J.A. 918

Ex. 43: Advertisement from *The Wall Street Journal*, dated August 5, 1992.....J.A. 919

Ex. 505: Advertisement from *The Wall Street Journal*, dated August 13, 1992 .....J.A.920

Ex. 507: Text of Jim Manzi's New York PC User's Group Speech .....J.A. 922

Ex. 508: Document entitled "Sales Speech" .....J.A. 923

Ex. 517: Key Reader portion of QUATTRO.MU file.J.A. 931

Exhibits From Brief of Plaintiff/Appellee from Court of Appeals for the First Circuit

Ex. A: Screen Shots (1-2-3, Quattro, Quattro Pro)...J.A. 969

Ex. B: Screen Shot (VP Planner +).....J.A. 978

PAGE

Exhibits From Briefs of Defendant/Appellant from Court of Appeals for the First Circuit

Lotus 1-2-3 Screen Display When Setting the Left Margin for Printing .....J.A. 979

Borland's Quattro Pro Screen Display in "123-Compatible Mode" When Setting the Left Margin for Printing .....J.A. 980

Lotus 123 Screen Display, Showing Available Command Choices and Long Prompt Message..J.A. 981

Borland Key Reader, with Blank Box .....J.A. 982

Microsoft's Excel Screen Display, with Lotus Menus .J.A. 983

Macintosh Screen Display .....J.A. 984

Statute

1976 Copyright Act, 17 U.S.C. § 107 .....J.A. 985

PAGE

The following items have been omitted in printing this joint appendix because they appear on the following pages in the appendix to the printed Petition for Certiorari:

Opinion of the United States Court of Appeals for the First Circuit in *Lotus Development Corporation v. Borland International, Inc.*, dated March 9, 1995. 1a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 12, 1993, as amended August 19, 1993 ... 29a

Permanent Injunction entered by United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated August 19, 1993 ..... 69a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated June 30, 1993, as amended August 19, 1993 ..... 71a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated July 31, 1992 ..... 106a

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Borland International, Inc.*, dated March 20, 1992..... 145a

PAGE

Opinion of the United States District Court for the District of Massachusetts in *Lotus Development Corporation v. Paperback Software International*, dated June 28, 1990 ..... 183a

The 1976 Copyright Act, 17 U.S.C. §§ 101, 102(a), 102(b), 103, 117 ..... 270a

Affidavit of Mitchell D. Kapor, dated November 27, 1989, submitted in *Lotus Development Corporation v. Borland International, Inc.* and *Lotus Development Corporation v. Paperback Software International* 281a

J.A. 540

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

VOLUME 1

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DEPOSITION UPON ORAL
EXAMINATION
OF
ADAM BOSWORTH

CONTAINS SPECIALLY CONFIDENTIAL MATERIAL
ATTORNEY EYES ONLY

10:02 a.m.
April 29, 1991
5400 Columbia Center
Seattle, Washington

Mary A. Hall, CSR
Court Reporter

[4] BY MR. KONRAD:

Q. Would you please state your full name and address for the record.

A. Adam Bosworth. My address is 9324 Southeast 57th Street, Mercer Island, Washington 98040.

* * *

[5] Q. Thank you. By whom are you employed, sir?

A. Microsoft.

Q. For how long have you worked for Microsoft?

A. I joined Microsoft October 1st, 1989, so since then.

* * *

[12] Q. When did you stop working as an employee formally for Borland?

A. Approximately June of 1989.

Q. When did you start working for Borland?

A. Approximately October of 1985.

* * *

[43] Q. What was your first position with Borland?

A. My first position was to manage the development of a new spreadsheet product.

Q. Was that the product that became Quattro?

A. Yes.

* * *

[53] For how long at Borland was it your responsibility to be the manager for the development of the spreadsheet product that became Quattro?

A. Until approximately May of 1988.

Q. By which point Quattro 1.0 was being sold, correct?

A. Yes.

* * *

[58] Q. Do you have any sense as to what proportion of your time you spent between October '85 and May '88 on the project that became Quattro?

A. Yes.
Q. How much?
A. Ninety-nine.
Q. As best you can recall, when did Quattro first get published?
A. Published?
Q. Sold, released.
A. December of 1987.
Q. Prior to that point within Borland itself, excluding the Hungarian fellows, who else was working on Quattro?
A. In what capacity?
Q. In a development capacity, let's focus on.
A. What do you mean by development?
Q. Let's say developers other than quality assurance people.
A. Okay. Inga Mortenson.
Q. Inga Mortenson.
A. David Chach. I believe it's C H A C H; **[59]** Charles Batterman, B A T T E R M A N; Peter Iaria, I A R I A; Tanj Bennet, T A N J, B E N N E T; myself, Adam Bosworth; Roger Schafly, S C H A F L Y. In a development capacity?
Q. Yes.
A. I believe that's all other than the Hungarians.
Q. And to be clear, I meant to exclude quality assurance people.
A. Yes, I excluded. These are only people that worked on code other than quality assurance.
Q. I appreciate that. With respect to this project before the time Quattro was released, did you have someone to whom you reported?
A. Yes.
Q. Who was that?
A. It changed.
Q. Initially to whom did you report?
A. Philippe, Philippe Kahn.
Q. And he was the chief executive of Borland?
A. Yes.

Q. Did there come a time when you reported to someone else?
A. Yes.

* * *

[64] Q. Among the people that you have identified who worked on the development of Quattro before it was released, is there one or more whom you would identify as the designer of its user interface?
A. Yes.
Q. Who is that?
A. Me.
Q. When we're using the term user interface there, how do you understand that term?
A. The totality of the ways in which the user can interact with the program, including the display, keys, the sequence of commands, the use of function keys, the navigation path to execute a process and the functionality that is displayed to the user.

* * *

[71] Q. To your recollection, who developed the 123.RSC file?
A. To my recollection, final responsibility on **[72]** that lay with Joanne Lin.

* * *

[74] Q. Did there come a time when it was decided that Quattro should have the ability to read and execute macros written originally in 1-2-3?
A. Yes.
Q. To the best of your recollection, when?
A. Between October and December 1987—No. Sorry. 1985, October, November 1985.
Q. During the first few months that you were there?
A. Yes.
Q. Who made that decision?
A. I did.

Q. Did Philippe Kahn approve that decision?

A. No. Excuse me. What do you mean by approve?

Q. Was there a moment when you went to Philippe Kahn who was your superior at the time and said I think we ought to do this, do you approve, and he said yes?

A. No.

* * *

[82] Q. Did there come a time in the development of Quattro when it was decided that the product should provide macro compatibility with 1-2-3?

A. Yes.

Q. When was that?

A. I don't know exactly.

Q. Do you recall approximately when?

A. I can only tell you by which time I know we had decided. There was no concrete point we said yes, we must be compatible.

Q. I understand. As best you can.

A. June 1986.

* * *

[83] Q. Who had made the decision that Quattro should have macro compatibility?

A. Me.

* * *

[88] Q. Did you consider it necessary in order to sell Quattro that you provide the macro compatibility with 1-2-3?

[89] A. At that point I was not sure.

Q. Did there come a time when you determined in your own thinking that it was necessary to provide macro compatibility with 1-2-3 in order to have market success with Quattro?

A. No.

Q. Have you ever reached that conclusion?

A. No. It could excel.

Q. I don't want to flog the subject—this could well be my last question on it—but the time you made that decision to have some macro compatibility, did you believe that providing that would at least enhance the market appeal of Quattro?

A. At which time?

Q. Prior to the time in June '86 when the gentlemen came over.

A. No.

Q. Did you conclude that at some later point?

A. Yes.

Q. Can you recall particularly when?

A. Not particularly.

Q. Was there anything further that you learned or studied or analyzed that helped you reach that conclusion?

A. Yes.

[90] Q. What was that?

A. Talking to expert spreadsheet users.

Q. Are there any in particular that you're thinking of?

A. The Cobb Group.

Q. Where were they located?

A. In Lexington, Kentucky, I believe.

Q. Among other things, Mr. Cobb wrote a very famous and successful book about 1-2-3?

A. Yes.

Q. That's the same fellow?

A. Well, it wasn't the same fellow I was talking to but he's part of that group.

Q. Fair enough. Do you recall with whom you spoke there?

A. I spoke briefly to his brother, I believe, and I spoke to Ted Cunningham or Tom Cunningham, the director of sales for that group.

Q. Do you recall what they told you about providing macro compatibility?

A. Not in the exact words.

Q. Generally. I understand.

A. Generally they felt it was important to be able to run all of 1-2-3 macros.

Q. Do you recall approximately when you had [91] that discussion?

A. No. There was numerous discussions.

Q. But this was after the discussion we were talking about before?

A. I believe so.

* * *

Q. Well, for what reason did the Cobb Group believe it was important to run all 1-2-3 Macros?

A. To enhance market success. That's different than achieving it.

Q. Fair enough.

Was there anyone else that you can recall specifically discussing that subject with?

A. There was a product 1-2-3 macro that was used for teaching 1-2-3 macro, teaching 1-2-3. I don't recall the name but I worked with the authors. And they also felt strongly that 100 percent [92] compatibility was important with the 1-2-3 Macros. And I don't recall their names but there were various beta sites who felt similarly.

Q. I'm just going to try some names. Was the macro product 101 Macros?

A. Yes. Thank you.

Q. When you say beta sites, beta is usually a term for testing outside a company after a product is well in development and coming up towards release, isn't it?

A. That's correct.

Q. As Quattro initially went into beta testing, did it not provide 100 percent macro compatibility?

A. No, it did not.

Q. Do you recall approximately when Quattro went into beta?

A. No.

Q. Can you peg it backwards from the time of release in November, December '87?

A. Only very approximately.

Q. Just so we can continue to talk, do you have any sense?

A. As approximately as—as precisely as I can remember, it was in the late spring or early [93] summer of '87. But I want to emphasize I'm not at all confident on those dates.

Q. Precision on that date is not my point here.

What kind of macro compatibility, if any, did the product have at that point?

A. With other products than itself?

Q. Specifically 1-2-3.

A. Limited.

* * *

[96] Q. You mentioned Joanne Lin. Do you recall anyone else who worked on the 123.RSC file from the time that the product went into beta until it was commercially released?

A. What do you mean by worked on?

Q. Developed.

A. What do you mean by developed? Do you understand what it takes to develop it?

Q. What do you mean by develop?

MR. GREGORIAN: Wait a minute. If he doesn't understand your question, you have an obligation to put to him a question that he does understand.

MR. KONRAD: I would like to do so by using terms that he's comfortable with.

MR. GREGORIAN: Develop is contextual.

A. By develop, I normally mean that you wrote code.

Q. Did new code have to be written from the time of the beta test on?

A. For what purpose?

Q. To implement the 1-2-3.RSC file.

A. Yes, but it's not part of the 123.RSC file.

Q. Is there a name for the file in which the new code was written?

[97] A. Yes.

Q. What's that?

A. Quattro.EXE.

Q. As the product went into beta testing as you recall, could the user cause the program to display 1-2-3 command readings?

A. I just don't remember exactly. I don't think so but I don't remember.

Q. As released, the product does allow the user to display 1-2-3 command menus, doesn't it?

A. It allows the user to display a command, allows the user to pick a menu which has words in common with the 1-2-3 menu.

Q. In Quattro, what file causes those words in common to appear?

A. 1/2/3.RSC.

Q. Are there any subdivisions within that file?

A. Any subdivisions?

Q. Are there separately identified components of the 123.RSC file?

A. Yes.

Q. Is there a separate component of it that causes menu display?

A. There is a component of the RSC file that **[98]** describes the menus to be displayed, yes.

Q. And that component contains within it the words that appear in the menu, right?

A. Yes, it does.

Q. Do you know who wrote that file?

A. The file is built by programming. The person doesn't write the file. They interact with the program to build it. I don't know if I'm making myself clear but—it's like asking who wrote a WK1 file.

Q. Was it written using the menu builder?

A. Yes.

Q. Do you know who created that file using the menu builder?

A. Someone working for JoAnne Lin. Whether it was her directly or someone reporting to her, I don't know.

Q. Do you recall any names of the people who worked for her at that time?

A. Uh-huh.

Q. Who?

A. Asha Nelson; someone called Cheryl, and I don't recall Cheryl's last name. No, I would recall more names if I were given them but I don't recall them.

[99] Q. To your understanding, did someone have to write a new file or using the menu builder create a new file that contained the menu commands from 1-2-3 in order to complete the 123.RSC file as it finally shipped?

A. Yes.

Q. Do you have any understanding as to whether the 1-2-3 software was used in that process?

A. No. I don't believe it was. I don't have a good understanding of that.

Q. Do you have any understanding as to whether documentation from 1-2-3 was used in that process?

A. I don't believe it was.

Q. Do you have any reason for not believing that it was?

A. Yes.

Q. What's that?

A. There was a book that was widely used by people in Quattro using 1-2-3 that was usually used by people worrying about these issues and I believe they used that book. Excuse me. It was either using 1-2-3 or tips and Techniques on—I don't remember the exact name of the book but there was a book that was widely regarded as being sort of the definitive statement of this.

[100] Q. Definitive statement of what?

A. Of what the 1-2-3 command sequence would be.

Q. After Quattro went into beta testing, you received information from beta testers, perhaps others, about having 100 percent macro compatibility, I think you described it. Was there some decision made at Borland to change the product and provide that?

A. We didn't have to change the product. There was a decision made to include the 123.RSC but virtually no lines of code had to be changed.

Q. Was the decision made to include the 123.RSC file in a way that would cause to display the 123.RSC menu tree?

A. Where? In the book?

Q. In the program when you're running it.

A. No. The product out of the box doesn't display the 1-2-3 menu tree. A decision was made not to do that.

Q. In order to cause the 123.RSC menu tree to appear as a user, you can load the program typing Q 1-2-3, can you not?

A. That's correct.

Q. And then you will see the 123.RSC menu **[101]** tree, correct?

A. Correct.

Q. And that is based on Lotus 1-2-3 menu tree, correct?

MR. GREGORIAN: Object to the form, based on.

Q. Where does it come from?

A. It is designed to be one that is compatible with Lotus's command sequence. What it means is that it would run Macros compatibly.

Q. It will also cause the program to display on the screen a sequence of menus using words from 1-2-3?

A. That's true.

Q. Now, was that understood when the decision was made when the product was in beta?

A. Yes.

Q. Prior to that, did the product have that ability?

A. The product for quite some time before beta had the ability to display any menu the user chose.

Q. But did it have preset a menu tree comparable to the 123.RSC menu tree?

A. That's not the product. There's a menu that looks—that is a superset of the 1-2-3 **[102]** command sequence but it has nothing to do with the product, at least the way I define product.

Q. I don't want to bog down on that difference at the moment.

MR. GREGORIAN: Well, the witness views it as significant.

Q. Well, I'm going to try to get at the meaning a different way.

As the product went into beta—

MR. GREGORIAN: Wait for the next question.

Q. As the product went into beta, did Quattro contain a preset menu tree like the 123.RSC menu tree?

A. I don't recall. I don't believe. As it went into beta, I do not believe it did, no.

Q. When the decision was made to include the 123.RSC menu tree as the product finally shipped, it was understood that the 123.RSC file would display a menu tree like that?

MR. GREGORIAN: Like what?

MR. KONRAD: Like the 123.RSC menu tree that it displayed when it finally shipped.

MR. GREGORIAN: Well, you either got a totality or you got a question. I don't understand. I object.

[103] Q. You knew when you were making the decision that you were going to display menu trees that had the words in the order from 1-2-3?

MR. GREGORIAN: Object to the form. That's not his testimony.

Q. Isn't that right?

MR. GREGORIAN: Object to the form.

A. I knew that in order to be 1-2-3 macro compatible, the simplest solution by far was to write a menu tree that insofar as its commands were compatible with Lotus was in the order the Lotus's was in, and at least began with the same first letters that Lotus's began with, and the easiest technology to be macro compatible by far at that point in the product was to implement such a tree. And, yes, one of the corollaries of that

is that the user will then see that menu instead of the normal Quattro menu.

MR. KONRAD: Could I hear the answer back.

(Record read as request.)

Q. As best as you can recall, that decision occurred in 1987?

A. Yes.

Q. And as best as you can recall, whatever work was necessary to turn the product from what it had been where it didn't have that ability in order to **[104]** give it the ability to display menus that had 1-2-3 compatible menu commands in them occurred in 1987, too?

A. No.

MR. GREGORIAN: Object to the form.

A. The product did not need to be converted in order to do that.

Q. Whatever work it took to allow the product—

MR. GREGORIAN: Could we try and save some time here? If instead of referencing back to your client's product, you would just ask him in terms of whatever work it took to display the l/2/3.RSC tree, because when you refer back to Lotus commands, you invariably generate an objection. I think what you want to know is when was the work done that was required to implement in the product the l/2/3.RSC tree and if you don't muck up the question with reference back to Lotus commands, I think you will get that answer and you can go on to whatever else you want to ask.

MR. KONRAD: I hear you and all I really want to get is when was that work done. However, it is a little hard for me to take that objection seriously when I am looking at a chart from the **[105]** Quattro reference guide which says "Menu items with square bullet indicate Quattro extensions to Lotus 1-2-3 menus." So whatever carefulness you are using now, whoever wrote this didn't seemed to be quite so persnickety about it.

MR. GREGORIAN: Whoever wrote it—

MR. KONRAD: Did it on behalf of Borland, I'm sure.

MR. GREGORIAN: —wasn't in a lawsuit and wasn't subject to having lawyers misconstrue it.

Q. When was the work done to permit Quattro to display the 1-2-3 menu tree?

MR. GREGORIAN: Object to the form.

Q. The 123.RSC menu tree.

A. Thank you. There was no specific work done with that 123.RSC in mind except in a very small amount during beta; one, no specific work done with 123.RSC in mind. I don't know how to say that better than that. There was work done to support allowing anyone to build any .RSC and that's where the work went.

MR. GREGORIAN: He's asking you a time. Maybe I can help. He's asking you a timing question. He wants to know when the work that Joanne Lin and Al did, whatever work it involved.

[106] A. Is that what you want to know?

MR. GREGORIAN: To—

Q. Now, I want to make sure I have this straight. When the product went into beta, it had some limited version of the 123.RSC file which did not display the 123.RSC menu tree in full, at least, correct?

A. I believe so. This wasn't a big deal to me.

Q. Somebody had to do some work to allow that to happen later on?

A. But not programmers.

MR. GREGORIAN: He is not asking who. He's asking when.

A. It wasn't under my control. I know when it was during the beta process.

Q. In '87?

A. Yes.

Q. What kind of work would it have taken?

A. Filling in the menu tree, using the menu builder and a very small amount of programming.

Q. How would you have used the menu builder to fill in the menu tree, as you put it?

A. Same way as you use the menu builder to build any menu tree in Quattro.

[107] Q. Describe how you would have created this 123.RSC tree using the menu builder.

A. Can I do that?

I would run menu builder from within Quattro. I would then, for each menu item in the first pop-up, describe the words that I wanted to see and either the associate action to take, if any, or the menu to go to, in this case virtually all of them ut one, go to a menu. Then, I would go menu by menu and do exactly the same thing until I was done. It's a boring and a tedious job, but it's not programming.

Q. I think what you have been trying to make clear to me is that the tools for allowing the menu items to perform associated actions were already in the product.

A. Yes.

Q. You had already defined through the menu equivalent commands, a set of actions which was sufficient in scale to look into this 1/2/3.RSC menu tree?

A. Virtually that's correct.

Q. I understand.

A. Okay.

Q. Now what you're telling me, though, is someone sat there and said main menu first item **[108]** worksheet?

A. That's correct.

Q. And then gave it the associated action?

A. Well, in this case they then added a menu.

Q. And go to another menu?

A. Yes.

Q. And then it says Range and did the same things and all throughout the menu tree?

A. Yes, that's what I'm telling you.

Q. Could a person have done that using just Quattro at that time?

A. Yes.

Q. I guess what I mean is where would they have gotten the words from.

A. Oh, I told you I thought they used that book.

Q. The intent was, though, to make sure that the menus insofar as they were the same as in 1-2-3 would appear in the same order and arrangement, isn't it?

A. The same order and arrangement is necessary just for macro compatibility.

Q. That was one of the implications of having the 100 percent macro compatibility that people talked about?

[109] A. Absolutely.

Q. And that became a design goal for the product?

A. Yes.

Q. Now we're looking here at the 123.RSC menu tree from this Quattro Reference Guide. We'll mark it in a minute. Do you know who created this diagram?

A. No. I remember it being drawn but I don't remember who did it.

Q. Did you check it? Did you review it?

A. No.

Q. Do you know of anyone who did?

A. I'm sure testing did it.

Q. To your understanding, they test it against the 1-2-3 menu tree?

A. I don't know how they tested it.

Q. During that beta period when the decision was made to go to that extent of macro compatibility, did Borland, to your knowledge, do any marketing research or studies, something of a formal nature?

A. No.

Q. Do you know of anyone else maybe that you weren't directly involved with that had provided some advice or thinking on the subject?

MR. GREGORIAN: Apart from what he has [110] already testified?

Q. Yes. I am asking for hearsay now. Did you hear anything from anyone else?

(Discussion off the record.)

A. The answer to that question is no.

Q. Who made that decision during the beta process?

A. Which decision?

Q. The one to include the 123.RSC menu tree in a way that displayed the menu commands.

A. I told you, once we decided to be macro compatible, that was the easiest mechanism for implementing it. I made the decision to be macro compatible and that was the easiest and most cost-effective way I could think of for making that happen.

Q. Did you submit that decision to Mr. Silverberg or Mr. Kahn for review?

A. Not that I can recall.

Q. Had that been a possibility that was previously discussed—

A. Yes.

Q. —within Borland?

Do you recall when?

A. I believe—and this is a very approximate [111] date—it was around April of 1987.

Q. Do you recall particularly how the subject had come up at that time?

A. Yes.

Q. How?

A. There was a suit by Lotus against VP-Planner.

Q. How did that affect you?

A. I told Phillipe that it would be possible to run 1-2-3 Macros quite easily without it having to be an integral part of of the product.

* * *

[113] Do you recall whether Mr. Kahn had any response to what you told him?

A. To the best of my recollection, the agreement that was made was we would make sure that the product in no way had the 1-2-3 command sequences as an intrinsic part of the product and we would deal subsequently with the issue of whether to expose a 123.RSC at all later.

Q. Now, later you decided to expose the 123.RSC menu structure, correct?

A. That's correct.

Q. Did you ever discuss that later decision with Mr. Kahn?

A. In what context?

Q. At all.

A. Yes.

Q. How?

A. We were talking about what should and shouldn't be in the product.

Q. Can you expand on what that conversation involved?

A. There was a question about which file [114] translators we needed, which graphic printers we needed and which RSCs we should include.

Q. Do you recall around when that occurred?

A. It was very late in the game. To the best of my recollection, it was on the order of September, October of '87. It may have been earlier but that's as close as I can come.

Q. Was there anyone else involved in this discussion?

A. I believe Brad Silverberg and Joanne Lin were involved.

Q. On the particular subject of which RSCs to include, do you recall Mr. Kahn's point of view if any?

A. No, actually I don't. I recall the discussion but not what his point of view was.

Q. Do you recall any particular disagreement on that subject?

A. Which RSCs to include?

Q. Yes.

A. Yes.

Q. What was the disagreement?

A. There was discussion about a SuperCalc RSC.
Q. Did you have one developed?
A. We had one in development.

* * *

[121] than the one that came before it, than the products that had existed heretofore, and any menu in technology will inevitably have its own implications upon the user interface because any menuing technology requires different alternatives based on its limitations. And in my judgment the optimal layout of commands given the menuing technology that I chose for Quattro—and I explored many—was the one that the Quattro menu embodies. So, to me, this 123.RSC sequence is fully not optimal. It was merely a migration tool to get over people's fear of flying, if you will.

Secondly, I had a profound belief that a product should never hard code the user interface that it exposes where it can be reasonably avoided based on my prior development, and Quattro embodied my attempts to make that possible. So, to me, what was integral to the product was the generic technology that allowed the product to function against a variety of different interfaces, including language translations and including customized interfaces by customers. That was what was integral. Some particular interface that either we chose to export or someone else chose to export, to me, is not integral to the product unless that interface optimally reflects the technology **[122]** that's used and is documented and is explained in the help system, and that was the Quattro RSC. That's what I mean.

Q. I even think I understand. No insult to you but I'm trying to follow along.

How difficult would it be to take out the 123.RSC file from Quattro?

A. To take it out?
Q. Yes.
A. How difficult would it be?
Q. Yes.

A. You mean how hard would it be not to include it in the product?

Q. What would it take to take it out?

MR. GREGORIAN: Physically?

A. As a physical file, you would simply not include it on the disks.

Q. So it would simply be to compile the product for release but leave out that file?

A. That's exactly correct. Not one line of code would have to be changed.

Q. Would that affect any other operation of Quattro?
A. Yes.
Q. What?
[123] A. 1-2-3 macro compatibility.

Q. But with respect to the Quattro menu tree or the execution of the menu equivalent command or any of the other components?

A. Not in the slightest.

Q. Just couldn't run 1-2-3 Macros or see the 123.RSC menu tree?

A. That's correct.

Q. Was that any part of your thinking at the time, that this was something you could get rid of quickly?

A. Yes.

Q. Why was that a factor?

A. In case there was some sort of legal issue. It was clear that it would be easy to change the product to not include that sequence or to not include it as an exposed RSC file.

Q. Forgive me for suggesting if it's too personal, but it sounds like you take a certain amount of pride in the Quattro menu tree.

A. Yes.
Q. Do you consider yourself the designer of it?
A. Yes.
Q. Do you think it's better than the 1-2-3 **[124]** menu tree?
A. Much.

Q. It's different, isn't it?

A. Yes.

Q. In fact, because of the menu build—Well, you helped create the menu builder, too?

A. I built it.

Q. And through that, any user can create their own menu tree that will tap into the same functionality, right?

A. Yes.

* * *

[125] Q. You don't feel the 123.RSC menu tree is an optimal solution to designing a menu tree that would effectively exploit the functionality that Quattro provides?

A. I don't think it's optimal for new users or serious users. I don't know if that's responsive.

Q. No, I understand. What you are saying is it might be optimal for people who are entirely familiar with 1-2-3?

A. In the beginning, it might be.

Q. When they come to Quattro first, this might be the better way for them to approach it?

A. That's correct.

Q. But the expectation is that they will move into the Quattro menu tree?

A. That's correct.

Q. That's what you hoped, at least?

A. That's correct.

Q. Should we take a quick break now?

(Recess.)

[126] Q. Before Quattro was commercially released, did you have any sense as to how many people you expected to use the 123.RSC file as opposed to the Quattro file?

A. Yes.

Q. What did you expect?

A. I expected that over time less than 20 percent would use it.

Q. Use the 123.RSC file?

A. That's correct.

Q. How about at the beginning?

A. Even in the beginning I expected a majority to use the Quattro menu tree, a substantive majority, on the order of 60 to 70 percent.

Q. Did others within Borland have different views?

A. Yes.

Q. What do you recall?

A. Well, people talking to the press giving different opinions and were relaying them to me. I don't know if they necessarily agreed with what the press were saying but they were relaying different opinions. The person who had the most different opinion was Rob Dickerson.

Q. What was his opinion?

[127] A. I don't know what percentages he would have put on it. He didn't agree that a vast majority would use the Quattro menu.

* * *

[128] Q. Do you recall whether the people that you spoke to, the people who expressed an opinion about having 100 percent macro compatibility also expressed such opinion as to what proportion of Quattro purchasers would use the respective menu trees?

A. Yes.

Q. What do you recall in that regard?

A. I recall some of the people we spoke to thought that a majority would use 123.RSC. Some of the people thought a large majority would use 123.RSC.

Q. Do you recall any of those who thought a large majority would?

A. I had my recollection refreshed yesterday and I now recall one of them.

Q. Is that Cobb?

A. No.

Q. Who?

A. Jared Taylor.

* * *

[140] Q. Can you give me an example of how they would indicate that?

A. By way of example, macros that were invoked using Alt in a character could be considered to be 1-2-3 macros.

Q. If you did that, then you could use another menu structure and another macro language but you just couldn't use the Alt character indicator to begin that macro?

A. Yes and no.

Q. How no?

A. In Quattro, macros include macro debugging. I don't know how you would support macro debugging without a menu tree that conformed to the macro interpreter's expectations. So, yes, you could do it but it would feel broken.

Q. Let me make sure I understand that. If you didn't want to provide a macro debugger, you could do what I suggest?

A. I'm not sure what it means, provide macros without a macro debugger. If you mean if I want to provide macros without a macro debugger, the answer is yes.

Q. Well, the macro debugger perhaps needs more explanation. What do you mean by that?

[141] A. Macro debugger is the ability to test and fix logic bugs in the process that the macro describes, without which, it is in my opinion, virtually impossible to write reliable macros.

Q. Let me see if I can put it into a more ordinary—if not, let me know—but using that hard coded interpreter and not having the 1-2-3 menu tree somewhere would mean that the only way that the user could work with the 1-2-3 macros was simply to execute them cold but they couldn't fix them, work on them, modify them. Is that what you're saying?

A. I wouldn't call it the 1-2-3 menu tree. But other than that, I would agree with that statement. I would just say generically if I provided a macro interpreter without debugging, the implications would be that you could run the macros but not understand what went wrong when something went wrong.

Q. You couldn't fix them, you couldn't modify?

A. Not without great difficulty.

Q. So I take it that back in that exploratory decision in '85 you determined not to follow alternative—

A. '86.

Q. I think this is the October to December '85.

* * *

[144] Q. What do you mean?

A. I mean Lotus.

Q. What do you mean?

A. I mean as I understand it, Lotus has announced a new product where they said the standard interface is now 1-2-3 keystroke compatible but they will be 1-2-3 macro compatible so I assume there is some macro technology underway.

Q. What product are you referring to?

A. There was a press announcement about 1-2-3W.

Q. If you hard coded the interface into the product including all the keystroke information, could you provide the ability for the user to essentially turn a switch and start typing on the keyboard and have the program respond as if you were using the product whose interface you had hard coded?

A. It's conceptually possible to do anything in programming. It's just a question of effort. It depends on how much time and how much space you want to take.

Q. I guess I'm trying to get at the difference between having the interface hard coded into the program and in a display. Would it be be possible to provide the ability for a user to invoke all the same [145] keystroke commands but not display the menus and the like—

A. It would be possible. It could cause cognitive dissonance.

D I S S O N A N C E. Cognitive is C O G N I T I V E.

Q. What do you mean by that?

A. What I mean by that is you would get no feedback into what you were doing and you would get completely lost and not understand where you were within the product.

Q. You being the user of the program?

A. Yes, yes.

Q. And that's because the display of the information in the user interface helps give you the context for where you are in the product?

A. You bet.

Q. And conveys information to you about interacting with the program, doesn't it?

A. You bet.

Q. Well, bear with me. I was just scanning through my notes to try to find where we touched on it before, but if we're overlapping, excuse me. I just want to come at it now having explored these points in more detail.

[146] What was the difference between the alternatives that we've just discussed and the alternative you discussed with Mr. Kohn around April of '87 in terms of how to achieve macro compatibility?

A. I don't know that it was April. I don't remember my comment. I don't recall it being around April. Do you mean —and I'm not quite sure which discussion you mean.

Q. You made reference to one time when, after the Lotus against Paperback suit, you mentioned to Philippe Kahn that there was a way to provide macro compatibility without displaying the menu tree.

A. That's right.

Q. How is that or how is what you were discussing there different from what we have just gone through?

A. It's different. The—

Q. Could you tell me how?

A. Once you have the technology that's in Quattro I—I don't know about Pro—which is essentially user interface independent technology, it's possible to tell the program to temporarily switch to an alternate interface, execute against that interface and switch back. The problem, as I've mentioned before, is you can't completely suppress **[147]** that interface because of debugging but you can make the interface one which is not exposed for the purpose of direct interface with the end user, and that solution I just described

uses the user interface independent technology rather than the hard coded user interface technology or the hard code standard interpreter technology.

Q. Now, when you use the term user interface independent technology—

A. That's correct.

Q. —what do you mean by that?

A. I mean that in large subset of what I earlier defined as user interface, it's described by the RSC file and that subset can be completely changed at will by building a different RSC, RSC file without affecting the product.

Q. And in your view Quattro was an example of user interface independent technology?

A. Quattro what?

Q. Quattro as released Quattro I.

A. No. we shipped an interface. It wasn't user interface independent. It was just that the code didn't depend on that interface. That was the ideal interface for the code.

Q. Although to some extent because of the Menu **[148]** Builder you did provide the ability for users to treat the product as if it were user interface independent?

A. Yes, it is user interface independent. I just don't think it's correct to say that Quattro was shipped being user interface independent. It did ship with interface and the documentation is very much dependent upon that interface.

Q. Do you recall whether there was ever any consideration given prior to the release of Quattro to document the 123.RSC menu tree?

A. There was some consideration given.

Q. Do you recall who considered that?

A. I don't recall the details but I do recall Mr. Dickerson considering it.

Q. Did someone decide not to?

A. Yes.

Q. Who?

A. Me.

Q. Why not?

A. In my opinion it was not a good interface. It was not the interface that I wanted users to use because I didn't believe it was the ideal or even a comfortable interface for most users using Quattro, and I did not want our imprimatur upon the user interface.

[182] IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

No. 90-11662-K
Civil Action
VOLUME 2

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.
Defendant.

DEPOSITION UPON ORAL EXAMINATION
OF ADAM BOSWORTH

SPECIALLY CONFIDENTIAL MATERIAL
ATTORNEY EYES ONLY

9:55 a.m.
April 30, 1991
5400 Columbia Center
Seattle, Washington

Mary A. Hall, CSR,
Court Reporter

* * *

[191] WITNESS: ADAM BOSWORTH 4-30-91 (VOL. 2)

For example, again using the idea of copying from a source block to a destination block, if one had different words for source block and destination block.

Q. You mean command words?

A. I mean words that would appear up on the top or the bottom of the screen but the order of the prompt and the semantics of the prompt were the same, then the instructions in the RSC file would be identical in terms of the order and the semantics to take. They would only differ elsewhere in a different part of the RSC file as to the exact text to display at that point in time. The reason this is important is this describes a fundamental behavior on the part of the product as opposed to any more trivial issue such as the exact words. And even the fundamental behavior I am saying is not part of the product intrinsically. It is described by the RSC file and the 123.RSC file would have to describe in totality to Quattro the fundamental behavior to take quite independently of what words it chose to display.

Q. I think I finally got it. Let me try it a different way and if I didn't, don't worry about it.

To some extent it sounds like what you are saying is that the 123.Alt file would not be a mode [192] that you would want a user to try to operate exclusively within because it may have been that not every single part of the Quattro functionality was enabled in that since it was really just known as a way to provide Macros and coming back to the Quattro menu tree?

A. That's my recollection. Now we had some internationalization issues with the Alt file late in the game and I don't remember precisely what ended up in the ALT file and what did not.

Q. Going back to an answer you gave me before, you said only the first letter is necessary to display to permit Macro execution. Why then does Quattro in the 123.RSC and ALT files display the full word?

A. In the ALT, I don't know that it does. I can take your word for it if you would like. In the RSC the goal was to kill

two birds with one stone and with one piece of technology. One goal was to allow for 100 percent macro emulation and for that only the first letter would be required.

The other goal was to make it possible for users who were familiar and comfortable with a different command sequence to migrate to Quattro without having to learn a new command sequence immediately, and for that second goal one wants to have [193] fully spelled out words.

Thirdly, I have alluded to debugging, and to debug a command sequence it is highly desirable to have as much information in the commands that appear while you're debugging as possible and that alone I think would be sufficient justification for fully spelling out the words at least in the commands. It might not be for the long prompts.

Q. Focusing just on those commands, both for the purposes of debugging and for how long users who were familiar with the 1-2-3 menu commands to migrate to Quattro, displaying full words in the menus gives them more information, makes it easier for them to use the product, right?

A. Than what?

Q. Than if all you used was the first letters which was all you needed for Macro execution, technically speaking?

A. No.

Q. Why not?

A. The ALT file, at least to my recollection, is not used as a primary interface.

Q. We're talking about the RSC I thought. You said you didn't know about the ALT we're talking about?

[194] A. For the RSC file yes, sir. If one is going to use an interface with commands, one wants the command as easy as possible.

Q. And having a full word is more helpful than just having first letter, right?

A. Yes.

Q. And just to pin it down, it's more helpful because it gives more information, right?

MR. GREGORIAN: The words give more information than the first letter?

MR. KONRAD: Yes.

A. Yes. That's not the only reason it's more helpful. It's one reason it's more helpful.

Q. Why else is it more helpful?

A. It's also more helpful because it's a reference to remember where you were in some command sequence.

Q. It helps you stay in context of where you are in the command sequence or menu structure, right?

A. Yes. Maybe that's what you meant by the first point.

* * *

[215] Q. Why then did you not try to encourage that right away by not including the 123.RSC menu tree at all?

A. How does that encourage it?

Q. Why not sell it by saying we have a menu tree, use it, convert now.

A. That doesn't encourage it.

Q. There is nothing in Quattro that will automatically convert your 1-2-3 macros into Quattro macros, correct?

A. That's correct.

Q. You could do something like that, couldn't you?

A. No.

Q. Excel does something like that, doesn't it?

A. It doesn't work.

Q. It works somewhat?

A. Can I answer this question, if I buy a car that runs 90 percent of the time and ten percent of the time I'm on the highway and it shuts down, I don't say to myself the car works 90 percent of the time. I say to myself the car doesn't work. It's that simple.

[216] Q. Have you analyzed the section on which Excel translates the 1-2-3 macros?

A. No, I have not.

Q. So 90 percent of the time, ten percent of the time analogy is not based on any estimation you have made based on the percentage of macros. That—

A. It's not made on any percentages that macro can translate of Excel.

Q. It's just numbers that you threw out for analogy?

A. No, that's not correct.

Q. Where do you get the 90 percent number?

A. I looked into translation.

Q. Of Excel?

A. No.

Q. For purposes of Quattro, you mean?

A. Yes.

Q. When was that?

A. I don't remember exactly.

Q. Do you know which macros from 1-2-3 Excel cannot automatically translate?

A. No, I don't know.

Q. Do you know any of the conditions in such macros that impede the Alt translation?

A. Certainly.

[217] Q. Which one?

A. Any self modifying macro.

Q. What percentage of macros would you say include self-modifying macro conditions?

A. Ten percent.

Q. What's that ten percent based upon?

A. My experience with writing, looking at and talking to customers, writing macros.

Q. And you would say you're an experienced macro writer?

A. No, I don't think I would say I was an experienced macro writer.

Q. But you are technically an experienced person?

A. I am a technically experienced person, yes.

Q. And the customers you talk to, are these the same experienced spreadsheet users you mentioned yesterday?

A. Yes.

Q. Did you talk to any novice users about the problems of translating self-modifying macros or the extent to which they use self-modifying macros?

A. No.

Q. Would you expect a novice user to use a self modifying macro?

* * *

[256] Q. I understand, but what might have happened is that you you would create something like the 123.RSC file. It would never be included in the product but just to make sure that the macro compatibility would work.

A. That might have been happened.

Q. But it seemed you had to find some solution of that type in order to test the Menu Builder?

A. Yes.

MR. KONRAD: I ask the reporter to mark as Exhibit 10 a one page-document bearing the production number 558088.

(Marked Deposition Exhibit 10.)

Q. Now, if you take had a look at this, does this appear to be something you recognize?

A. I have seen this before.

Q. Do you recognize this as a piece of electronic mail that you received?

A. Yes.

Q. You'll see in the first sentence it refers to the 6/8 version of MT. Is MT Mark Twain?

A. It would appear I was mistaken about the sequencing of Palm Springs and Mark Twain.

[257] Q. Or it's also possible the person was using the old code name. Do you recall at this point?

A. I don't recall. I just know we had both versions. I don't recall the sequence points.

Q. Whichever sequence this came in, they're the same product?

A. Uh-huh.

Q. Fair enough. The memo says in the second paragraph: "I like the new menu's look much better. It's a much cleaner screen look." Do you have an understanding as to what "new menu's look" is referring to here?

A. Yes.

Q. What is that?

A. Three things. Number one, the menu, not that much earlier had automatically displayed sets which made them wide and they had been fixed to display narrow unless they asked to wide settings.

Two, at some point not that long before there, they had displayed on the left side of the screen, not the right side of the screen, and they had been moved to the right side of the screen.

Q. That's where they would appear when they popped up, as you called it then?

A. Right. And three, we are constantly **[258]** thinking we are tuning the actual sequence of commands to be included in the Quattro interfacing. This was yet another go-around.

Q. When you say Quattro interface there, you specifically mean the Quattro menu commands from the Quattro RSC files?

A. I mean specifically Quattro RSC.

Q. Right. Now, this memo appears to contain some discussion about where commands should be in the menu tree, how they should be placed. Do you recall much discussion with Mr. Silverberg on that point?

A. I already testified I had discussion with everybody on these points.

Q. And there was back and forth as to where these things should go?

A. Absolutely.

Q. Any idea how many such discussions?

A. Not precisely.

Q. Too many to count?

A. On the order of a hundred.

Q. Any idea how long it took to finally settle upon a Quattro menu tree?

A. To finally settle upon it?

Q. Yes.

A. Yes.

[259] Q. How long?

A. Approximately a year.

Q. Any idea how much work it took? By any measure of work, you can give it man-hours, days.

A. It wasn't that much work. It was just—

Q. Just a lot of thinking?

A. No. It was just lot of iterations. Man-hours, I wouldn't—

Q. Iterations is kind of a computer term but it means a lot of different tries, a lot of different experiments to see which was better?

A. Getting a lot of different people's judgment. I don't know exactly how much work on it.

Q. But over a period of time various communications occurred and eventually it was settled upon?

A. Yes.

Q. The third paragraph, I guess, says: "While it's nice to make the top menu alphabetic, it doesn't work so well in practice." Had the top menu of the Quattro menu tree been in alphabetic form up to that point?

A. Yes.

Q. As shipped, it's not quite, is it?

A. No.

* * *

[265] Q. I understand but now you have had a chance to see this again.

A. I don't remember the precise timing any better seeing this.

Q. You have no reason to believe that the 123.RSC file started before July 4th?

A. No, that's not true.

Q. It may have been done?

A. Yes.

Q. But it wasn't done by then? At least that's what this memo suggests.

A. It's what this memo could suggest, yes.

Q. Do you think it was done by July?

A. I don't remember.

Q. But do you recall working with Joanne Sakai Lin on the 1-2-3 menu tree following this memo?

A. I recall working with her on it. I don't recall the timing.

Q. What do you recall doing with her?

A. She would call me up and say I need that command to do such and such and I don't see where this command is in the Menu Builder. And then I would either tell her or I would say whoops, you are right. This is one one case where we would have to add one **[266]** and I would add a command. There were a few such instances.

Q. These were commands from 1-2-3 that she couldn't find an analog in the menu equivalent commands for?

A. Yes.

Q. So you would have to create a new menu equivalent command to address that?

A. Yes.

Q. Do you recall any specific such commands?

A. No.

Q. Do you recall approximately how often that came up?

A. There were fewer than ten, is my recollection.

Q. When she'd call up, would she describe the command by the word from 1-2-3, I don't see anything that's equivalent to File Range, whatever?

A. My recollection is that most of them had to do with the status menu. They weren't really commands.

Q. But you recall this coming up with some real commands?

A. Printing. I don't remember the details.

* * *

[301] A. He was very vague. He wanted to build a great spreadsheet. He wanted to have great charting. He wanted to have great math stuff, minimum recalc. That was about it.

Q. What you have described, did that occur in one conversation or more than one conversation?

A. There were essentially three conversations.

Q. During any of those conversations at or about the time you joined the company, did Mr. Kahn indicate that one of his objectives was to create a Lotus clone?

A. No.

Q. Did he indicate that one of his objectives was to create a product that looked just like the Lotus product to the user?

A. No.

Q. At any time at or after you joined the company, was it your objective to create a Lotus clone?

A. Absolute—

MR. KONRAD: Sorry. I object to the form.

A. Absolutely not.

Q. At any time after you joined the company and began working on the spreadsheet product, project, was it your objective to produce a product that would **[302]** appear very much like Lotus to the user?

MR. KONRAD: Object to the form.

A. I'm not sure what you mean by that question.

Q. At any point was it your objective to create a spreadsheet product for Borland that had a user interface that was identical to the Lotus interface?

A. Absolutely not.

MR. KONRAD: Object to the form.

Q. In terms of your own goals and objectives for the spreadsheet project, what were they?

MR. KONRAD: Ask for clarification. At what time are you talking about?

Q. Let me rephrase it. At or about the time you joined Borland, did you establish some objectives or goals for the spreadsheet project?

A. Yes, I did.

Q. Can you tell us what those were?

A. Yes. My key objective was to use the Borland languages as an extendibility mechanism for the applications with the product being envisioned as the first in a family of applications. A secondary objective was to have a superset of the charting capabilities of Harvard presentation graphics so that **[303]** users could do charting, real charting, without having

to buy a different product and then import data from one product into another.

Q. Any others?

A. There were numerous others but those were the two key objectives. The other were basically just to do a bunch of things much better for the spreadsheet user than any product had done before and that included macro recording, macro debugging. It included a better menuing system. It included extensibility things like functions. At the time it was intended to include multidimensional worksheets. Those were my objectives.

Q. Were some or all of those objectives that show in the Quattro product that was released in 1987?

MR. KONRAD: Object to the form. I think he said Quattro problems.

A. No.

MR. KONRAD: In any event, I do object to the form.

Q. Let me restate the question. Were any of those objectives achieved in the Quattro product that was released in late 1987?

A. Many of them were achieved technically.

* * *

[328] Q. Do you recall when in time you reached the conclusion that a pop-up menu presentation was the one you wanted to use in Quattro?

A. Yes.

Q. When was that?

A. It was approximately February of 1986.

Q. Do you recall when in time you began designing the Quattro menu tree?

A. Yes.

Q. When was that?

A. It was approximately February of 1987.

Q. Was there a reason that you began the Quattro menu tree design as late as you did?

A. Yes.

Q. What was the reason?

A. In my experience building products, it's unproductive and fruitless to worry about the sequence of commands until you understand both the presentation of the commands and the set of commands that will be available. And any attempt to try and finalize that command sequence before that is a pure waste of time **[329]** and even is misleading to people such as documentation, so I make it a policy not to define or worry about command sequences until such time as the presentation and the functionality are either finalized or close to finalized.

Q. And you did that in this particular instance?

A. Yes, and I did that in this particular instance. There is a second reason on this particular case.

Q. Okay.

A. I don't want to get into a debate with people about the exact words. There was an experience I went through at Analytica which I referred to as the UI Wars where extraordinary vituperative and vicious internecine warfare would go on about the particular order of a word or even a particular word in question, which was one of the most unproductive spectacles I had ever witnessed in my life and I wanted no part of it.

I didn't want, therefore, to design the Quattro command sequence until such time as the Menu Builder was done. And, for example, Philippe gave me a hard time—and in fact this actually happened—said here's a Menu Builder. Go design one that's **[330]** better and come back and we'll talk about it and finalization of both long prompts and commands to someone who is a better wordsmith than myself and I needed the Menu Builder in order for that to be possible.

Q. The point being, I take it, that once you had the Menu Builder it was relatively easy to change the command terms.

A. It was trivial except for the implications upon documentation.

Q. But if, for example, Mr. Kahn didn't like your choice of the word Copy, it was trivial to use the Menu Builder to change it, to duplicate or something like that?

A. Exactly.

Q. Now, I take it that what you have just described was not the only reason that the Menu Builder was written.

A. It may not have been the only reason but it was probably the primary reason.

Q. Did the Menu Builder have anything to do with the user customization capability of the product?

A. Yes, it did. As I said before, I believe that users are heterogenous in nature and their commands will vary. Also, I wanted a way for users to **[331]** splice both add-ins and macros into a user tree so a user could deliver to another user an application that seemingly integrated into the menu tree and the menu tree allowed me to deliver, deliver that functionality. Also, it allowed me to deliver the tree needed for 1-2-3 macro compatibility. So there were a variety of reasons but I would say the main reason was to stay out of the U.I. Wars.

Q. When did you design the Menu Builder?

A. To the best of my recollection, between January and March of 1987 is when I actually designed and wrote the Menu Builder.

Q. Is it fair to say that at that point it had not been decided that the product would incorporate what eventually became the 123.RSC menu tree?

A. It's fair to say we hadn't even thought about it one way or the other.

Q. My point being that you did not set out to build the Menu Builder so that you could incorporate in the product what eventually became the 123.RSC menu tree.

A. That's correct, I did not.

Q. And the Menu Builder had other purposes and was used for other purposes besides that, correct?

A. Yes.

[332] Q. And those are the purposes you described?

A. Yes, to allow for civilized dialogue about the menus that we ship.

Q. Did it improve the quality of the dialogue or did it just allow you to say if you don't like it, fix it?

A. No, it actually improved the quality of the dialogue.

Q. Now, you had given some testimony yesterday, the substance of which as I understood it was that the menu technology chosen dictates or constrains the eventual menu organization. Do you recall that testimony?

A. I recall the testimony.

MR. KONRAD: I object. I think it mischaracterizes his testimony. I don't remember him using the words dictate or constrain.

Q. Let me substitute the word everything. Do you recall testimony yesterday to the effect that the menu technology affects the optimal menu structure in organization?

A. What I recall saying, though perhaps not in these precise words, was that the presentation of the menu has a major effect of the choices one makes for the layout in grouping of commands.

* * *

[338] A. I think it would be more correct to say I wasn't the principal designer or principal author of the long prompts.

Q. But you were the principal author of the command terms?

A. Well, in my own mind I was. I don't know that everyone else would have agreed.

Q. And how about this structure or the sequence and organization of the menu, were you the principal designer of that?

A. Yes, I was.

Q. With regard to your efforts in designing the Quattro RSC command terms and menu hierarchy, did you have some specific objectives or goals in mind that you were seeking to achieve by that design?

A. Yes, I did.

Q. What were they?

A. Well, first was for clarity. One is to have the terms be as self-evident as possible to as large an audience as possible; second, was shallowness of hierarchy. These are not necessarily in any priority of purpose as much as just goals, meaning that in general it would be ideal to take as few keystrokes as possible to reach the desired [339] command; third, which can conflict with that sometimes was to ensure that the more common a command the fewer keystrokes it took; and fourth was to logically group the commands for the purpose of showing logically grouped state or settings. Those were, I would say, the four key objectives I had.

Q. Is it fair to say that overall your objective was to design a menu tree that was as functional as possible?

A. Certainly.

MR. KONRAD: I object to the form.

A. Yes.

Q. And in executing the design, you attempted to achieve the four specific objectives that you just identified?

A. Well, I and others attempted to achieve it. It's a very pragmatic process.

Q. Is it fair to say based on your previous testimony that the objective of logically grouping the commands flowed from the decision to display associated states?

A. Not quite. It's fair—

* * *

[347] Q. Mr. Bosworth, I'm showing you a document that was marked this morning as Plaintiff's Exhibit 10. This was the electronic mail from Mr. Silverberg. I take it Mr. Silverberg was one of a number of people who reviewed and commented on your menu tree design.

A. That's correct.

Q. He indicates in the first paragraph of this memorandum that some of the trade-off in this cleaner screen look is that he has to hunt for some commands that are, quote, buried a menu or two down, close quote. Do you see that?

A. Yes, I do.

Q. That, I take it, refers to the issue of a flat hierarchy versus one that is deeper.

A. You have actually refreshed my memory on something.

Q. Okay.

A. It does but not quite in the sense that I think you mean it. Originally—

Q. How does it differ?

[348] A. Originally I had a flatter hierarchy and a longer set of menus and you had to go down less frequently but you had more on the order of 13 or 14 items per menu and it was strongly felt by Philippe and some others, and really myself after I thought about it, that that was too many, that the menu was intimidating. And so the menu tree that the new menus included—I forgot one of the things the new menus included—was actually a deepening of the hierarchy to some higher degree. In the goal of a hierarchy, we had really gone too far.

Q. The result being that some menus contained, in your view, too many command terms?

A. Well, particularly I have to say—in all honesty I'd have to say in Brad and Philippe's view.

Q. I just want to understand the adjustment that was made. I take it the adjustment was reduce the number of command terms per menu and make the hierarchy a little deeper or a little less flat.

A. Right. It was a touchy trade-off.

Q. Why was it touchy?

A. Well, the more experienced a user, the fewer the keystrokes you want the user to have to type to accomplish a given piece of functionality, particularly if he is of high frequency. At the same [349] time, you don't want to completely intimidate and terrify the less experienced user and you don't want to obscure a larger fraction of the screen. So there is a trade-off there, as there usually is in user interfaces, between what's best for the novice and what's best for the expert.

Q. Is it fair to say generally that these criteria or guidelines or rules that you applied in designing the Quattro menu tree sometimes point in different directions?

A. Absolutely.

Q. And that that gives rise to trade-offs; that is, to achieve one objective, to satisfy one criteria, you have to sacrifice the other to some degree?

A. Absolutely.

Q. Is what we just talked about an example of that?

A. Absolutely.

Q. I take it that reasonable people can disagree on which rule ought to be sacrificed in favor of which other rule. Is that a fair statement?

A. Yes.

Q. And that happened in this case, that is, there were disagreements or discussions along these [350] lines?

A. There were discussions in this case.

Q. You had indicated before lunch that you began work on the design of the Quattro menu tree in February of '87. When did you finish?

A. I'm not sure I really finished until close to ship.

Q. Any idea what the total time commitment on your part was to that aspect of the project?

A. Probably overall about ten working days.

Q. Is it fair to say that notwithstanding what you've testified to earlier about user interface wars there never came a time when you handed the design over to somebody else and said just go do it? In other words, you were the principal designer from start to finish, I guess is my question.

A. That would be fair to say with respect to organization of the commands and the wording of the commands. It wouldn't be fair to say with respect to the long prompts.

Q. I understood from a previous answer that you would not regard yourself as the principal designer on the long prompts, is that correct?

A. That's correct.

* * *

[358] Q. Why did you regard the ability to debug Lotus 1-2-3 macros as important as at the time Quattro was released?

A. In my experience, no programmer can properly maintain a test or guarantee the quality of their programs without some debugging strategy, and typically by far the most productive strategy in my experience is to have a debugger. I regarded then and I regard now Lotus macros as a programming language since they were used in the ways I'm familiar with programming language as being used, and as such I saw exactly the same need for these debugging tools, and every discussion I ever had with any customer who wrote and/or tried to debug Lotus macros only confirmed my opinion in this regard that the single biggest productivity gain I could give to people who were writing Lotus macros was a debugger.

Q. Now, you were asked yesterday about some alternative technologies, alternatives to the 123.RSC resource that was included in Quattro. And one of those that you identified in response to Mr. Konrad's questions was a hard coded interpreter. Do you recall that testimony?

A. Yes, I do.

Q. Now, had that solution been adopted by [359] Borland in lieu of what they did choose to do, what would have been the effect on the ability to debug Lotus macros?

A. It would have made it more difficult.

Q. How so?

A. When you are debugging a Lotus macro, not infrequently you have a command sequence which you expect to behave in one way and isn't behaving in that way because of a logical mistake on your behalf. And the only way to really understand that is to see the command sequence play out against the backdrop of the menu tree and understand where you made this logical mistake. Of a hard coded interpreter, there would be no such hard coded menu tree to play out against and therefore you couldn't display to the user at all what happens, what would then appear to be relatively mystical. Keystrokes were fed into the maw, M A W, of this interpreter.

Q. So in that situation any attempt to debug would be flying blind, so to speak. You could not step through the menu tree and determine where the logical error was. Is that what you're saying?

A. Any attempt to debug where viewing the menu tree would help with the solution, in my opinion then and now, is a significant percentage of the time would [360] be flying blind.

Q. Another alternative that was discussed in the testimony yesterday was hiding the menu, I think were your words. Do you recall that?

A. Yes.

Q. Would that have basically the same effect as you just described on the ability to debug macros?

A. Yes and no.

Q. Tell me how yes and how no.

A. The hidden alternative I described was one where there was an alternate resource file but it wasn't visible. In that case you actually could display the command sequence but the user would have no way to test it outside of the macro. There is a better debugging strategy than the one that we just described but it's still a much worse debugging than the one where the user can test fragments of the macro by hand using the same command menu tree that the macro is going to use and then debug it. So this is a less pernicious alternative but still limited.

Q. The third alternative that was discussed in the testimony yesterday was a hard coded interface that would be 1-2-3 macro compatible.

A. That's correct.

Q. What would the effect of that choice have [361] been on the ability to debug macros?

A. Well, the effect of that choice on the ability to debug macros would have been final, I mean the user would have been able to see one menu sequence and debug against it. A minor subnote is that that's the only interface that would have been available to the product and—

Q. Why would that be?

A. Well, if you write the code to hard code user interface, that's your user interface code. You are not going to write two totally different pieces of interface code. It's ridiculously expensive. Also, hard code user interface is a very foolish strategy in general but it doesn't necessarily adversely affect the macro debugging, per se.

Q. When you referred to that alternative yesterday and testified about it at some length, I take it you had in mind what you just described, that hard coded interface would be the only user interface offered, is that correct?

A. Yes.

Q. Did you also have in mind that under that alternative the Lotus menu tree would appear on the screen?

* * *

[369] BY MR. GREGORIAN:

Q. Mr. Bosworth, you were asked some questions this morning by Mr. Konrad concerning the portions of 123.RSC that were necessary to support macro execution and you indicated in that regard that one of the portions that was not necessary was all but the first letter of the command terms. Do you recall that testimony?

A. Yes, I do.

Q. What effect would including a 123.RSC only the first letter of the Lotus command terms have had on macro debugging?

A. Only including the first letter would have meant that when they were debugging the macro, you would see a very, very skinny pop-up menu consisting of single first letters, and for most letters you would be pretty unlikely to have a good idea of exactly what each letter alluded to. And so again it would complicate the job of debugging the macro and make it less felicitous.

Q. In that situation, I take it it would have **[370]** been likely that the hypothetical person attempting to debug a macro would have had to at least on occasion go to some other course to figure out what the first letter stood for.

A. It would sort of be analogous to going back from 1-2-3 before they had words. Yes, it would be analogous. It would be possible. It would just be more clumsy.

Q. Just so we can define for Mr. Konrad the scope of what he believes he's been deprived of, how many meetings did you have with counsel for Borland at which the subject matter of a Lotus claim or the legality of releasing Quattro with the 123.SRC file was discussed?

A. While I was at Borland?

Q. Yes.

A. I can recall two.

Q. And I think you indicated yesterday that one or more of those conversations was with an outside law firm or representative of an outside law firm. Was it one or was it more than one?

A. That's a good question. I honestly don't recall if it was one or both now that I think about it.

* * *

[389] Q. But what I mean to ask you is whether there is any technical constraint, in theory at least, that you know of which would prevent you from designing a program that displayed the same menu tree that's in Lotus 1-2-3 but using pop-up menus instead of ring style menus.

A. The identical menus, right?

Q. Yes.

A. No.

Q. Did the use of pop-up menus for the Quattro menu tree affect the arrangement of the commands within any given menu?

MR. GREGORIAN: I'm sorry. Can I just hear the question.

(Record read as requested.)

A. Yes.

Q. How so?

A. There is a logic to showing associated state that would dictate that even within associated state, some state should be

coupled with another state and that would dictate a particular order even within a single menu.

Q. Now, I believe you have said several times something to the effect that you don't believe that there's any one user interface that would be [390] necessarily ideal for all users. Have I caught the gist of something you said?

A. You have caught exactly what I have said.

Q. Why is that?

A. I believe based on extensive experience with customers that customers' needs vary just as individuals vary. Customers' needs are heterogeneous and therefore the relative frequency with which a customer may perform an action almost varies from customer to customer. Any interface which is predicated upon the concept that making things that are most frequent most accessible and the least frequent the least accessible therefore has to vary from customer to customer.

Q. Do you think that some user interfaces have more appeal to some individuals and others have more appeal to others?

MR. GREGORIAN: Object to the form. I don't know what appeal means.

Q. Is the only respect in which different users might prefer different user interfaces the frequency with which they would use different commands?

MR. GREGORIAN: Look, to simplify this, apart from the appeal issue, you are not talking about [391] user interfaces, which he has defined very broadly. I think you mean menu tree. Maybe you don't but you're kind of switching gears on him.

MR. KONRAD: Well, actually, I don't mean to. I asked the question user interface. He mentioned with respect to commands.

Q. That's why I'm trying to now explore whether or not the only way in which you think that different user interfaces might have different suitabilities for different users has to do with the frequency with which they use particular commands.

A. I didn't mention menu trees. I mentioned commands. I would like to distinguish. That includes, for example, what the optimal function layout key is. The—no, that's not the only consideration.

Q. Well, do you think that some users just simply like one menu tree better than another?

A. I don't know what that means.

Q. Do you think that some menu trees might strike some users as more intuitive and logical whereas another user would have had another organization to think it would be more intuitive and logical?

A. Yes.

[392] Q. Do you think that some users might enjoy working with one menu tree more than with another?

A. Yes.

Q. Do you think some users might find one arrangement of a menu tree to be more informative to them as they're working with the program than another arrangement?

A. Yes.

Q. Do you recognize that there is some room for taste and personal appeal in designing a menu tree?

A. Yes.

Q. You told Mr. Gregorian that you don't know who wrote what we've been calling the long prompts in the 123.RSC file but you also told him you didn't think they were the same as in 1-2-3. Why do you think that?

MR. GREGORIAN: I'm sorry. Which file are we talking about?

MR. KONRAD: 123.RSC.

A. My recollection is that at one point I was asked to technically autoset the long prompts for the effect of picking the menu item was to go to a child menu, to the comma separated, child separated menu and I didn't do so. I thought it was a bad technology.

* * *

[396] Q. Do the menu trees convey information to the user?

A. Sure. That's one of their functions.

Q. Now, if you you recall, I objected at one point and I apologize for doing so but I cannot understand this term patterned usage that you kept using in one of your answers. You were talking about that in connection with the differences between the 123.RSC menu tree and the menu tree in the 1-2-3 product, and what I'd like to begin by asking you is what difference does the pattern of usage make, in your mind?

MR. GREGORIAN: Wait a minute. What difference to what?

Q. What difference to anything does the pattern of usage make? Why were you mentioning that term?

MR. GREGORIAN: Why did he include that in his answer that you argued gratuitously at the time?

MR. KONRAD: I object. I don't think that's fair.

MR. GREGORIAN: Why did you think that was so significant that you mentioned it?

* * *

[399] is going to buy a Quattro would have a need at some point to run 1-2-3 macros?

A. I really didn't know.

Q. Do you mean to testify that you believed the only way to sell Quattro to include 1-2-3 macro compatibility?

A. What do you mean by the only way to sell Quattro?

Q. Do you think you could have sold some units of Quattro without the 1-2-3 macro compatibility?

MR. GREGORIAN: Do you mean like more than one?

MR. KONRAD: Yes.

A. I think I could have sold more than one unit of Quattro without the 1-2-3 macro compatibility.

Q. Then was it the only way for Quattro to be commercially successful, in your view, to include the 1-2-3 macro compatibility?

A. That's an overstatement.

Q. What do you think?

A. I thought it was—would have been a large, unnecessary risk to the commercial viability to sell Quattro without 1-2-3 macro compatibility.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Case No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

v.

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF ROBERT HESTON DICKERSON, JR.

THURSDAY, SEPTEMBER 12, 1991

CONFIDENTIAL
SUBJECT TO PROTECTIVE ORDER

Reported by: Victoria Jensen, CSR# 6393

---

* * *

[4] Q. Would you please state your full name and address for the record.

A. Robert Heston Dickerson, Jr. I'm at 210 Carbonera Drive, Santa Cruz, California.

Q. And when were you born, sir?

A. January 17, 1957.

Q. By whom are you employed?

A. Borland International.

Q. What's your current position there?

A. I'm Vice President and General Manager of the Database Business Unit.

* * *

[9] Q. And in fact, when you joined Borland was responsibility for spreadsheet products included in what you did?

A. The product management responsibility for spreadsheet was included.

* * *

[15] Q. And how would you describe the change in your responsibilities at that point with respect to the product that became Quattro Pro?

[16] A. Before I managed someone else who took care of the product and focused strictly on the product management responsibilities. Afterwards, I took direct management responsibilities for the product, and did whatever needed doing, irrespective of the area in order to ship the product.

Q. At the time of this transition in early 1989 did you become more active in the development side?

A. Development? Describe "development side".

Q. Working more closely and more actively with people such as Mr. Warfield who are actually developing the product.

A. Yes, I did work closely with Bob Warfield.

* * *

[62] Q. Mr. Dickerson, at any point prior to the release of Quattro Pro do you recall participating in any discussions concerning the legality of including the 1-2-3.MU file in the product?

A. None that I can recall, no.

Q. And do you recall ever participating in any discussions about whether or not Lotus was going to sue or object to the 1-2-3.MU file?

A. That seems like the same question. The answer is no.

Q. Now I believe you testified before that you didn't particularly have an opinion as to whether it was necessary to have the 1-2-3 macro compatibility in the product for it to be commercially successful. Whatever you precisely said before will speak for itself, but I do recognize that we talked that. I mean to ask a slightly different question. Do you recall whether you had any view at the time that Quattro Pro came out as to how important it was to have macro compatibility with 1-2-3?

A. Yes. I did feel it was important to have macro compatibility with 1-2-3. I'm not quite sure how to measure that, but I felt that was important.

[63] Q. Were there any particular audiences or target markets, if you will, for whom you thought that would be particularly important?

A. Yes.

Q. What were those?

A. Those who would be people that already had Lotus 1-2-3.

* * *

[64] A. I think that any computer user is likely to be a Lotus 1-2-3 owner.

Q. Fair enough. I mean—

A. Lotus.

Q. What you're trying to tell me at that point you're not sure you would really distinguish that kind of customer as one that be more likely to be a 1-2-3 owner than anyone else.

A. If somebody has an 80 percent market share or thereabouts, it's hard to say. Anyone using a spreadsheet is likely to be a Lotus owner, irrespective if they're in a corporation or in a home.

Q. Do you recall whether up to the time that Quattro Pro came out, Quattro, the original version had been perceived as a successful product?

A. By me?

Q. Well, by you, I'll start with that and then ask if that was shared.

A. My sense was that the original Quattro was not as successful as we could be in the spreadsheet category.

Q. Did you have any views as to why that had been—

A. Yes. I felt that the product was buggy. It was not able to load very large spreadsheets, so if you were—if you had a copy of Lotus on the machine and a

* * *

[94] promote it to make people want it. We call it a green crystal because it's like Tide, why does Tide get your clothes whiter? They have green crystals and if someone asks what's in those green crystals? So our green crystals was VROOMM, which stood for Virtual Run Time Object Oriented Memory Manager.

And the thing that VROOMM allowed us to do, was load very large spreadsheets into Quattro. And so we were promoting that you have all these new features and not to have to move up to a bigger machine in order to load your spreadsheets.

Q. Well, was part of the compatibility without cost message related to users potential need to learn a new product?

A. Well, in our marketing materials it was a part of it.

Q. How did that play out?

A. Well, the factor was we wanted to go out to people using competitors' products and say, here's Quattro. It's easy to switch to. And so we promoted everything that we could think of that would make Quattro easier for Lotus users to adopt, if we were selling to a Lotus audience.

Q. Well, such as?

A. The things that we focused on were spreadsheet and macro compatibility, capacity, keystrokes, and there's **[95]** one more. Let's see. Well, I guess that's all, file compatible, macro compatibility and keystrokes.

Q. What do you mean by keystrokes?

A. The keys that you press when you use the product.

Q. In other words, if you used the 1-2-3 menu tree file at least in Quattro Pro you could press the same keystrokes that you already knew from 1-2-3?

A. That's correct.

Q. Is that what you mean by keystroke compatibility?

A. Correct.

Q. Would that same concept have been described in the messages as ease of learning? Or well, as ease of learning?

A. That's certainly possible, yes.

Q. All right. Do you recall any of the marketing trying to get at the concept by discussing re-training costs?

A. It seems I do, yes. I think I do remember that.

Q. Well, let me back up one step whether or not, because what I don't want to do is ask you to try to remember in your mind what the particular words were used in any piece of literature, but do you recall whether part

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION

vs.

BORLAND INTERNATIONAL, INC.

November 8, 1991

Oral deposition of JAMES C. EMERY, held in the offices of Advocate Court Reporting Services, Inc., 12th Floor, 1525 Locust Street, Philadelphia, Pennsylvania, 19102, commencing at 10:00 a.m. on the above date before Linda Luber, a Certified Reporter and a Notary Public in the Commonwealth of Pennsylvania.

---

* * *

[202] Q. What's more critical to the value of the software, the words that are coding for the functions or the functions themselves?

A. Well, before the learning is taking place and assuming a reasonable choice of words, then I would say the functionality.

But once a user associates a cluster of things or a specific function with a word, then it takes on additional value. It affects the market acceptability of the product.

But if you're saying—when you're first designing the menu, you have a huge scope of alternatives. So that there are out of the virtually infinite variety, there are a large number of structures that are perfectly adequate and it would be very difficult to say one is better than another.

Once you congeal the design and have successfully marketed it and spend the money **[203]** to get people to use it and they've trained on it, then the economics change.

You incurred a cost, both a marketing cost and a learning cost on the part of the users that change the economics.

Q. When that change occurs, does there ever come a time where the words chosen are more critical to the value of the software than the functions that they represent?

A. If you introduce the product where the functions are exactly the same, the menu structure was exactly the same, but all you did was arbitrarily make a choice of words, I think that would diminish the value of the product because users would then have to learn a new set of words.

In fact, it's conceivable that what they know before would have a negative effect, because now they have to unlearn what they knew and learn a new set of words.

Q. I'm going to move to strike. That is not what I asked you.

What I asked you, sir, was in this context that you're describing, whereas the product is released and the words chosen have **[204]** greater value or increasing value over time as users use them, my question is, does there ever come a time when the words chosen are more critical to the value of the software than the actual functions the software performs.

MR. KONRAD: I object. You move to strike. It's your right to do so. I think he tried to answer your question in the right

way. It's difficult to make comparisons except by complex reasoning. I think he was trying to answer your question before.

Do you understand the precise question?

THE WITNESS: No, I don't understand the question.

A. You're implying that as time moves on, the shifts in economics change and at some point something happens. I don't understand what you are saying.

As people learn—Let's say the market expands until more and more people learn the specific choice of words. It seems to me one could argue at some point—let's say the **[205]** functionality at the beginning was more important than the words, but relatively speaking, the words assume greater value as the population expands that knows it, and therefore, would be difficult to shift.

So that phenomenon can take place. I'm not sure if that's the phenomenon you're referring to.

Q. That's precisely the phenomenon I'm referring to.

Does there ever come a time where the words become more critical to the value of the software than the functions themselves?

MR. KONRAD: In all cases or can you posit a case?

BY MR. GREGORIAN:

Q. Can you posit a case where that would be true?

A. Sure. Particularly if the alternative is almost the same functionality. It's this complex trade-off that we've been talking about in several cases today.

* * *

[209]

* * *

Users have to weigh the enhanced functionality of the new product versus the learning. And the learning in part is learn-

ing new words. That's not all the learning but that would certainly be a part of it.

There could certainly well be a situation where new products do not offer sufficient enhanced functionality that a substantial number of users would be willing to switch.

So at that point you could say the value of the words exceed the value of the [210] functionality. The relative functionality.

You always have to compare things relative to alternatives.

* * *

[236]

* * *

Q. Doctor, you say in Paragraph 113 that whether alternatives could have been chosen at the beginning that would have done equally well is irrelevant, once users did learn the 1-2-3 menu commands, those particular commands acquired great significance. Their current value lies precisely in the familiarity that millions of 1-2-3 users have acquired with them; correct?

A. Correct.

Q. Is there anything that you believe has contributed to their current value besides this familiarity people have acquired over time?

MR. KONRAD: Is it fairer to say [237] that your question is when you said the current value line is precisely in the familiarity, does he mean it lies only in the familiarity?

MR. GREGORIAN: All right. That's a good way to put it.

MR. KONRAD: So he understands you. I don't know what the answer is.

A. No. It includes things like training material available, text books that are available, a population of trained people that you can hire to use this, a set third party developed products that are compatible with your product.

There tends to be a whole structure that grows up around the successful product and we get a positive feedback mech-

anism, that the value of a product increases greater value, which in turn, further increases the value through all these ways.

And Lotus was in the fortunate position of having all those contributive values.

BY MR. GREGORIAN:

Q. But I'm not asking about and I don't think your speaking in Paragraph 113 about the value of [238] the product. You're talking about the value of the menus and the menu structure; correct?

MR. KONRAD: I'm not sure that the answer is different from the question.

MR. GREGORIAN: That's what I'm trying to ascertain.

MR. KONRAD: I think the question, Professor, is whether or not these things you are speaking of concerning training and support and all of that are part of what you see in the value of the menu structure?

MR. GREGORIAN: Correct.

A. Well, the reason a user would be unwilling to change is that they would have to learn a new product. If they jump to a new product, they wouldn't simply be foregoing whatever advantages of the old product and learning that they invested in it, but a whole set of auxiliary advantages of third products, trained user population, textbooks, et cetera.

So it's a major shift when you move to another product that isn't compatible with all of the infrastructure that has grown up around [239] a successful product.

I don't think you can divorce the issue of that menu from all these other things. They're part and parcel of the same thing. It's part of the hurdle of moving to a different product.

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Volume I
Pages 1 to 176
Exhibits See Index

Civil Action
No. 87-0076-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

PAPERBACK SOFTWARE INTERNATIONAL, et al.,
Defendants.

Civil Action
No. 87-0074-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

MOSAIC SOFTWARE, INC.,
Defendant.

THIS TRANSCRIPT IS DEEMED CONFIDENTIAL
PURSUANT TO COURT ORDER

DEPOSITION OF BERNARD A. GALLER, a witness called on behalf of Paperback Software International, taken pursuant to the Federal Rules of Civil Procedure, before Ralph J. Simpson and Robin Gross, Registered Professional Reporters and Notaries Public in and for the Commonwealth of Massachusetts, at the Offices of Sherin and Lodgen, 100 Summer Street, Boston, Massachusetts, on Thursday, January 25, 1990 commencing at 12:50 p.m.

* * *

[116] Q. Did you receive any other written materials from Lotus?

A. I believe that's all.

Q. Did you receive any other written materials from Lotus' lawyers?

A. I will correct my statement. I believe everything I received came through the lawyers.

Q. Now, I see a reference here, Professor Galler, November 9, 1989, to review of outline. What is that outline, please?

A. That may have been an outline representing a draft of the first declaration.

Q. Who prepared that outline?

A. It was prepared at my direction, and after a number of conversations the first outline was prepared by counsel to reflect what I had told them. They very graciously offered to do the writing and the text processing to start with.

Q. Had you given them any written input prior to the preparation of that outline?

A. I believe I made available to them some of the testimony from the Digidyne/Data General case, **[117]** where I did largely a tutorial discussion on computers and computing, and operating systems and so on for that case. I believe I had made a copy of that available to them, along with a notebook with pictures of the visual aids I used in that tutorial in that case. I think that's all. Most of the input for that outline was our discussions.

Q. I show you two volumes, and I ask if this is the testimony to which you were referring.

(Pause)

A. Okay, I'd like to qualify or explain my previous answer.

Q. Please.

A. The answer to this is yes, I believe these are the transcripts of my trial testimony; one representing primarily my presentation as the tutorial, and the second day probably being almost entirely the cross-examination.

Now, what I said I made available to counsel here was not the transcript of this. Judge Orrick in that case required that all direct testimony be submitted to him in advance.

Q. In writing?

A. In writing.

[118] Q. Sounds familiar.

A. Therefore, my entire tutorial had been prepared in writing ahead of time. And I believe it was a copy of that submission, prepared ahead of time, that I made available to Mr. Beckerman. I think you will find, if there's a comparison of that document and the transcript, that it's very close. I followed it almost verbatim, but I won't—my style is not to read to the jury but to explain.

Q. So if I can just make sure, Professor Galler; I take it, then, that the volume I'm now holding is essentially your reading your prepared direct testimony, perhaps not word for word, and it was actually the written direct testimony which you had sent to your lawyers?

MR. BECKERMAN: I object to the word "reading."

A. I just said I didn't read it to them. I would say there's a close parallel because it reflected what I intended to say, and I had it in front of me, highlighted and so on to guide me in what I said, but it was close. I did not read it.

Now, further than that—let me see that again, because you say "this volume," and I don't **[119]** know if there's anything in here besides that.

MR. BECKERMAN: And let the record reflect the volume under discussion is the excerpted reporter's transcript of Tuesday, March 24, 1981, in Digidyne Corporation against Data General Corporation, et al.

Off the record.

(Discussion off the record)

Q. Professor Galler, I take it that the direct examination in the volumes which I handed you is substantially the same, although not identical, to the material that you just testified that you sent to Mr. Beckerman.

A. That was the intent, and I think it is. I note that in this first volume dated March 24, my tutorial discussion ended almost at the end, on Page 373, and I think cross started after that, so most of this volume is the tutorial.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS
Volume I
Pages 1-176
Exhibits 1-3

CIVIL ACTION
NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—VS.—

BORLAND INTERNATIONAL, INC.,
Defendant.

SPECIALLY CONFIDENTIAL,
SUBJECT TO PROTECTIVE ORDER

CONFIDENTIAL DEPOSITION OF BERNARD A. GALLER, Ph.D., a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Joan M. Kelley Cassidy, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Wednesday, August 28, 1991, commencing at 1:00 p.m.

PRESENT:

O'Sullivan, Graev & Karabell
(by Henry B. Gutman, Esq.)
30 Rockefeller Plaza, New York, NY 10112,
for the Plaintiff.

Fenwick & West
(by Edmond C. Gregorian, Esq., and
Stuart P. Meyer, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[41] Q. Any other litigation expert engagements with Lotus besides the Paperback case, as we have defined it, and this case?

A. No.

Q. Any consulting relationship with Lotus past or present—

A. No.

Q. —other than in connection with these cases?

A. No.

Q. Any discussion with any representative of Lotus about possible future work in the litigation area or consulting area?

A. No.

Q. Nobody at Lotus has told you that there might be more work of this nature for you in the future—

A. No.

Q. —that's what you are testifying?

A. That's correct, the answer is no.

Q. Now, with regard to the litigation consulting that I have been asking you about, how much income did you derive from that type of **[42]** activity in 1990?

A. I believe somewhere in the area of $50,000 to $60,000.

Q. How about 1989?

A. Perhaps the same.

Q. 1988?

A. I don't know. I don't make any attempt to remember those figures.

Q. And if I broadened the question to all types of consulting, including non-litigation related, what would be your income from consulting activity in 1990?

A. About the same. Most of my consulting in the last few years has been of this nature.

Q. Okay. And would that answer apply to '89 and '88 too; that is, the non-litigation consulting would not add a great deal to the figures you have been already given?

MR. GUTMAN: Wasn't '88 the year for which he didn't remember?

MR. GREGORIAN: Yes.

Q. But whatever the figure is, the question is, did the non-litigation consulting add significantly to X?

[43] MR. GUTMAN: I'll object to the form, but you can answer it.

Q. It's a little metaphysical, I agree.

A. I understand. Part of the question is what is significant. But I mentioned that most of my consulting was of this kind. My relationship with NCR on the committee I mentioned is an example of some other work that I do. But most of my consulting is of the litigation kind in recent years.

Q. "Most" is a little unspecific too.

A. It's as good as your "significant."

Q. Okay, maybe better. Would consulting outside the litigation sphere have added more than 10 percent to the figure you gave me for 1990, which was $50,000 to $60,000?

A. No.

Q. Would it have added more than 10 percent to the figure you gave me for '89, which was $50,000 to $60,000?

A. I don't think so.

Q. What was your salary from the University of Michigan in 1990?

A. About $98,000.

[44] Q. And how about 1989?

A. About $94,000.

Q. Now, you had indicated in response to some questions that you were asked in the Paperback case, that you had not used the 1-2-3 product before being retained by Lotus in the Paperback case; is that correct?

A. I believe so.

Q. You had also indicated—and I believe you were asked this question during the trial—that up until the time you testified at trial, you had used that product for approximately three hours; is that accurate?

A. I believe that's right.

Q. Since the Paperback trial, have you had occasion to use Lotus 1-2-3 some more or not?

A. Except for the experiments we already discussed—

Q. Today's experiments?

A. Yes.

Q. Okay.

A. Except for those, no.

* * *

[76] MR. GREGORIAN: Exhibit 2. (Document marked as Exhibit 2 for identification)

Q. Doctor, I am showing you Galler Exhibit 2 for identification. Can you identify that as a copy of your resume or curriculum vita?

A. (Witness reviews document) It is, as of some date. I believe in 1990.

Q. So it's current as of that date?

A. As of whatever date it was produced, yes. And it is fairly recent; but I have a more recent one, which I don't think is material here. I just want to be sure that it's understood this is—as of the date it was produced, this is correct.

Q. Now, calling your attention to the bibliography section which begins about three or four pages in—

A. Yes.

Q. —are there any publications since whatever date this was prepared?

A. Yes. I believe I have added one or two in for 1991.

Q. Do you recall what those are?

A. They are in the area of intelligent vehicle [77] highway systems. They are not relevant to this case.

Q. They have nothing to do with user interfaces or—

A. No.

Q. And where did those articles appear?

A. I think they're listed as technical reports at the University of Michigan.

Q. What does that mean, a technical report, in the context of the University of Michigan's business?

A. It means it's not an official refereed publication; it's a document which is distributed to people who might be interested in receiving it.

Q. Within the university community?

A. Well, and often to other universities or other interested parties.

Q. So these reports aren't published anywhere in any journal or—

MR. GUTMAN: I object to the form.

A. The reports represent an informal publication process that—the same content, sometimes even the same papers, are submitted for publication to formal journals; but this is an early [78] publication process. Some of them never do get published more formally.

Q. Have you submitted these papers for publication?

A. I don't think so.

Q. Do you have a co-author on these papers?

A. I think one of my students was a co-author on one or both of them.

Q. What was his or her name?

A. His name is Avi, A-v-i, Rubin, R-u-b-i-n.

*Q. All right. Calling your attention to the publications on the first two pages of the CV that we do have in front of us, which I believe takes us back to 1978, are there any of those that deal with user interface design or computer/human interaction topics?

A. Would you repeat that, please.

*(Question read)

MR. GUTMAN: I object to the form of the question.

A. Computer/human interaction is a very broad area. I mean, even these intelligent vehicle highway systems articles discuss computers in the car interacting with the driver, interacting with [79] central computers on the roadside; and those are computer/human interactions.

Q. Let me limit it, then, to user interface design. Are there any on the first two pages that deal with user interface design issues?

A. I don't think so.

Q. Are there any in the bibliography at all that deal with user interface design issues?

MR. GUTMAN: I object to the form.

A. On page 3 under 1973 there's a description—a paper listed which describes an interactive student registration system which reports the results of a student project which eventually developed into the student class registration system at our university.

Probably in that paper there's reference to the fact that we designed into the system a user interface for the operator of the terminal where the registration would be done. I don't think there is any real discussion of it. I think there is a mention of it.

Q. Okay. Any others that mention—that discuss user interface design besides the one you've identified?

[80] MR. GUTMAN: I object to the form.

A. I don't think so.

Q. Now, if I could call your attention back to the opening entries on the bibliography, there is, for example, in the first item, a parenthetical that says "with K. Chen." What does that mean, "with K. Chen"?

A. Well, Professor Kan, K-a-n, Chen was a co-author of that paper.

Q. Is Professor Chen the so-called first author of that paper?

A. I believe—

MR. GUTMAN: I object to the form.

A. I believe I was first author in the usual sense of one of those two, and he was the first author on the other.

Q. Okay. And "first author" in the usual sense means what?

A. Did most of the writing. In both cases, of course, the work was regarded as joint; but one of us did more of the actual writing of the paper.

Q. You believe that was you in one case and he in the other; is that correct?

A. Yes. I believe the first one would have **[81]** listed him as the first author, and the second one would have listed me.

Q. And in terms of the work that was done that led up to those papers, did he do most of the work, or did you do most of the work?

A. No. I think it was quite joint, quite equal.

Q. Equal division?

A. I think so.

Q. Equal as those things can be; is that right?

A. That's right.

Q. Is Chen a professor at Michigan?

A. Yes.

Q. Is he in the engineering department?

A. He is in my department, electrical engineering and computer science. He is in another division of the department.

Q. Now, there are some references here to what appear to be conferences or symposia. For example, the third item, "Thoughts on Software Engineering," refers to a conference on software engineering in Pittsburgh in 1989. Are those papers, or are those transcripts or summaries of remarks you made at the

[82] conference?

MR. GUTMAN: I object to the form.

A. Well, you are saying "those." Let me talk about that one—

Q. All right, talk about that one.

A. —because sometimes the circumstances are different for different ones. That particular one I was on a panel which was

discussing the progress of software engineering over the last 25 years; and each of us prepared a kind of position paper in advance, which was later published here, as indicated in this item.

Q. And this item indicates that you were—that your paper that you prepared in advance was one page?

A. One page.

Q. Now, the next item, "The History of the Computer, Part One. . .," *Humanities Education*—is that a publication—

A. Yes.

Q. —*Humanities Education*?

A. That's correct.

Q. And who is the audience for that publication, if you know?

[83] A. I believe scholars, primarily in the academic community in the humanities.

Q. And your contribution on this particular item was approximately five pages; is that right?

A. That's correct.

Q. Now, there is an entry under 1985 entitled, "An LSA Computer Environment." That's the title, I take it; is that correct?

A. That's correct.

Q. And then that's followed by, quote, *LSA*, the University of Michigan. Is that a reference to a publication?

A. Yes. There's a—I don't think it exists now, but at that time there was a magazine, a—something between a magazine and a newsletter put out by the College of Literature, Science, and the Arts for their alumni and faculty; and I wrote an article for them about the environment in the College of Literature, Science, and the Arts dealing with computers.

Q. I see, all right. 1986, the item that appeared in the IEEE International Conference on Systems, Man, and Cybernetics Proceedings—is that a paper that was prepared in advance, or is that **[84]** some sort of transcript or summary of remarks that you gave at the conference?

MR. GUTMAN: If you remember.

Q. If you remember.

A. I don't really recall too well. That arose out of a seminar that I headed—that I organized and chaired in Japan in 1985, and I believe I prepared a short report about that seminar and gave it at this international conference in Georgia. And, again, it was about a one-page summary about what had happened in Japan the previous year.

Q. I see. Is this bibliography for the period 1985 to the present and the two publications that you referred to from 1981 on here—is it complete?

MR. GUTMAN: I object to the form.

A. It's complete—I believe so.

Q. And your updated resume that you referred to—would that have the two '91 publications listed?

A. Yes. I think I have a copy of that in my briefcase, and I can give that to you tomorrow.

* * *

[89] Q. The courses this year, Computer Science 280—is that a semester course or a full-year course?

A. Semester course.

Q. And the IVHS course—is that a single semester or full year?

A. The lecture is one semester. We will have a laboratory part the second semester, so . . .

Q. Are you participating in both?

A. I'm—I have been participating—well, I mentioned my participation in the lecture part.

Q. Yes.

A. In the laboratory part—I have been supervising the organization of it. I won't actually teach it.

Q. Okay. And taking the period prior to 1989, the last advanced course—strike that. Let me ask you a better way. In 1988 what was your teaching—

A. That year was also the 181 course.

Q. The last advanced course, advanced computer course, that you taught at Michigan was in 1987; is **[90]** that correct?

A. I believe that's right.

Q. And the last intermediate course that you taught was in 1984; is that correct?

A. I believe that's correct.

Q. You have never—as I understand it at least, and correct me if I am wrong—you have never taught a course on user interface design; is that correct?

A. That's correct.

Q. In your own courses, you would spend perhaps an hour, but no more, on user interface design issues; is that accurate?

MR. GUTMAN: I object to the form.

A. I can't answer that because I don't usually segregate out such issues and identify them as such. In all of my teaching I mention, when appropriate, issues of design. And, for one thing, when I talk about design, that includes interface design. I do spend some time on design questions.

Q. Well, design questions aren't limited to user interface design questions, are they?

A. Not limited to, but they would cover them too.

* * *

[95] A. I have heard recently that there is some work in the psychology department. I don't know about courses, and I don't know specifically who does it.

Q. The human factors course—is that in your department?

A. No, it's industrial and operations engineering, No. 436.

Q. Is its subject matter limited to computers, or does it go beyond to other types of man/machine interactions, if you know?

A. I believe it's limited to computer type interactions.

Q. Do you know Professor Green?

A. Yes.

Q. Do you have professional contact with him?

A. Yes.

Q. Any knowledge or information concerning his competence in the field of human factors?

A. I believe he is recognized as quite competent.

Q. I'd like to ask you some questions about your involvement in software design. And because of the length of your career, I think we, for the sake [96] of efficiency, ought to limit it to the last ten years. Is it correct that in the last ten years the only applications software whose design you have been involved in is this project that you undertook with your son?

MR. GUTMAN: I object to the form.

A. Well, for one thing we haven't introduced the project that I did with my son. Are we supposed to all understand what that is?

Q. Well, I am going to ask you some questions about that; but I'm trying—you were asked about that—

A. We will agree that there was a project with my son.

MR. GUTMAN: He has the Paperback transcript, so. . .

Q. I don't want to reinvent the wheel here. I have read the transcript.

A. I understand, but I am not used to no foundation questions. But I will accept it.

MR. GREGORIAN: I can't ask him legal questions? He's objecting on foundation grounds.

* * *

[101] A. Yes, a product resulted. Unfortunately, I have not been able to convince everybody at the university who is involved to use it. I lament that because I think they're foolish. They are continuing to use a very primitive input method for the product—that the product they bought requires, and it's going to cost the university a great deal to continue to do it that way.

Q. Is the product that resulted from your student work called something? Does it have a name?

A. It was called CEDARS, C-E-D-A-R-S.

Q. Is that an acronym?

A. Yes. The product the university bought was DARS, D-A-R-S, where the "A" is "Audit"; but I don't remember the

rest of it. But the "C E" we put on the front is probably "Computer Environment" or something.

Q. Let's talk about the CEDARS program. That has not been placed in service at the present time, correct?

A. No. It was demonstrated. It worked. It has not been adopted.

Q. Technically speaking, is the CEDARS program [102] which you've described an applications program or a systems program?

MR. GUTMAN: I object to the form.

A. That question sometimes comes up. Is a FORTRAN translator part of an operating system? I generally include it. Other people sometimes don't. I think this would be in the same nature of that question.

Q. I.e., it's borderline?

MR. GUTMAN: I object to the form.

A. It's borderline. In my opinion it's part of an operating environment; but, I mean, I'd be just as happy, in the context of whatever the discussion is, to call it an applications program if that's useful.

Q. With regard to the other item you mentioned, small programs for your own use, can you give me an example of what you are talking about there?

A. Oh, programs for computer grades, programs for keeping track of resources, whatever.

Q. Now, with regard to—when you say, "Keeping track of resources," you mean like personal financial applications?

[103] A. That kind of thing and related things, yes.

Q. Did the CEDARS program have a user interface?

A. Yes.

Q. What was the nature of that? Was it menu based or command driven?

A. I guess I don't recall.

Q. Was the user interface for that program specified in your student assignment, or was it something the students designed?

A. The students designed, which is why I don't recall because I didn't specify it.

Q. The program—and, again, I just—if you recall a name, it would help in framing the questions. The program that you did with your son—did that have a name, either commercial name or acronym?

A. CORP, C-O-R-P.

Q. CORP, C-O-R-P?

A. Right. It was an acronym, but I don't remember now what it was an acronym for.

Q. Did the CORP program have a user interface?

A. Yes.

Q. What was the nature of it?

[104] A. It was demand driven.

Q. And who did the design work on that?

A. My son and I.

Q. Together?

A. Together.

Q. Equal division of effort on that or—

A. I think so. I remember now that it was run on a TRS 80 Model I, which puts it approximately in 1980, 1981.

Q. Do you recall how many different commands there were?

A. I don't recall. I think there was a fair amount of prompting from the system once a command was given. I don't think there were too many commands.

Q. More than ten?

A. Maybe about ten.

Q. Who designed the prompts?

A. I think we did most of the design equally.

Q. How long did the total project take?

A. I don't recall, a number of months. I mean, we were both part-time people.

* * *

[117] Q. Are there any texts or treatises in the field of user interface design that you regard as authoritative?

A. I don't know of any.

Q. Do you know of any texts, whether—or treatises, whether you regard them as authoritative or not?

A. I am sure there are some. I think there are some well known works by Ben Schneiderman. I think he has a book or maybe even more than one, but I have not used them. I have not read them.

Q. Have you read any texts or treatises in the field of user interface design?

A. No.

Q. Have you read any articles on user interface design?

A. I don't recall.

Q. Are there any articles on user interface design that you know about that are regarded in the field as authoritative?

A. Again, I have seen references to some. I may even have read one or two, now that I think of it, by Schneiderman; but I don't recall specifics.

* * *

[123] Q. Are you aware of the SIGCHI special interest group within that organization?

A. I am aware of it, yes.

Q. What is it, sir?

A. Special Interest Group on Computer/Human Interaction, I believe.

Q. Are you a member of that group?

A. No.

Q. Have you ever been a member of that group?

A. No, it didn't exist when I was president.

Q. Have you ever attended conferences sponsored by that group?

A. No.

MR. GUTMAN: Well, I'll object to the form. I think it's potentially misleading. Don't they hold joint conferences with the ACM?

MR. GREGORIAN: Look—

THE WITNESS: No. He said conference sponsored by them, and I haven't been.

MR. GUTMAN: Okay. Just to be fair, they sponsor some conferences jointly with the ACM. Do you mean to exclude those or not?

MR. GREGORIAN: Come on, you are putting questions to him now. If you want to cross-examine **[124]** him, that's fine.

THE WITNESS: I don't think I have been at any conferences in which they have been a sponsor.

Q. Have you been at any conferences at which they have been a joint sponsor with the ACM?

A. No.

Q. Okay.

A. That's what I meant.

Q. Have you ever read published conferences—conference proceedings of SIGCHI?

A. No.

Q. Are you a member of any other professional group or association or organization whose focus is user interface design?

MR. GUTMAN: Object to the form, lack of foundation.

A. I am not a member of any group which advertises its focus, whatever that means, on user interface design.

Q. Same question for computer/human interaction. Are you a member of any such organization or group?

MR. GUTMAN: Same objection.

A. No.

[125] Q. And just so we don't have a problem at some later time about what I mean by "focus," I mean a group whose primary interest or objective or raison d'etre is user interface design, as opposed to the multitude of other areas that I am sure are equally interesting in the computer field. That's what I mean.

MR. GUTMAN: I object to the form.

Q. Is your answer the same?

MR. GUTMAN: I object to the form of the question. You mean the same as the answer about not belonging to groups that advertise themselves that way?

MR. GREGORIAN: Correct.

A. Yes.

Q. You refer in your affidavit in this case to having conducted a review of Quattro and Quattro Pro in connection with your work on this matter. Can you describe for me physically what you did to review those products?

A. Well, it included this morning's work on the machine; but I—

Q. Well, you have described that in some detail. Let's set that aside. Let's say **[126]** preparatory to your declaration, you looked at the products, right?

A. I looked at the manuals—well, Quattro, I looked at the product. Quattro Pro I did not see the product and interact with it. I read the manuals.

Q. Okay, fine. I am glad I asked. With regard to Quattro, let's talk about your review of the product. You had it up on the computer and running, correct?

A. Yes.

Q. What physically did you do?

A. I am not sure I know what the question is.

Q. Well, your affidavit says that you reviewed it. I don't know if that means you looked at the opening screen, you worked through the menu hierarchy, you balanced your checkbook, you created a spreadsheet and made changes to it. I am just trying to understand in a general way what this review consisted of. That's all. It's not a trick question, I assure you—

MR. GUTMAN: Please.

Q. —as opposed to something else.

MR. GUTMAN: Oh, come on. I will object to **[127]** the form of the question.

MR. GREGORIAN: Okay.

MR. GUTMAN: Not to mention the incredible —

Q. Okay. It is a trick question. Answer it anyway. I just want to know what you did.

A. I installed it.

MR. GUTMAN: I object to the form of the question. Tell him what you did.

A. I installed the product during the period when I was installing several other products over the course of several weeks. In each case I exercised it to some extent. In some cases—and I don't remember if Quattro was one of them—I actually went through the tutorial that came with it. In others, I tried various combinations of things that one usually does with spreadsheets, in some cases created a back road to see how one did it and how one would invoke the macros and what was involved in working with the system.

I don't recall specifically how many of those various activities I did with any one of the products, such as Quattro.

Q. When did you do this activity?

[128] A. In preparation for the Paperback case.

Q. Oh, in the Paperback case?

A. Right.

Q. Since—when —

A. Oh, I'm sorry. I did bring that up again. I mean, I did bring up the screen and try a few things to refresh my memory for Quattro—oh, I'm sorry. I installed Quattro Pro—after the Paperback case?

MR. GUTMAN: Yes, the time frame is confusing.

A. Yes, I was thinking wrong. Before the declaration?

Q. Yes. That's what I am asking.

A. I'm sorry. I installed Quattro Pro as well, and I did the same kind of exercising on that. I just was thinking incorrectly.

Q. All right. Let's back up a step.

MR. GUTMAN: Let's clarify.

THE WITNESS: Yes.

MR. GUTMAN: What he was referring to before is what he did in the Paperback case.

A. That's right. I think I was thinking for that case.

[129] MR. GREGORIAN: I have this proprietary instinction that I like to ask the questions.

Q. Is what you were referring to before with regard to Quattro work that you did in the Paperback case?

A. Yes.

Q. All right. Now, between the time you did that and the signing of your declaration in this case—

A. Yes.

Q. —did you do anything else by way of running, looking at, exercising Quattro?

A. Yes.

Q. What did you do with Quattro? And I will get to Quattro Pro, but tell me Quattro first.

A. Yes. Because of the issues in this case, I brought it up again on the screen. I went through the exercise of going from one menu structure to the other, back again, and tried a few simple things. I I didn't feel I needed to do very much because I was familiar with it.

Then I installed Quattro Pro—

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Volume II
Pages 2-1 to 2-248
Exhibits 4-11

CIVIL ACTION
NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.,

Defendant.

SPECIALLY CONFIDENTIAL,
SUBJECT TO PROTECTIVE ORDER

CONTINUED CONFIDENTIAL DEPOSITION OF BERNARD A. GALLER, Ph.D., a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Joan M. Kelley Cassidy, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Thursday, August 29, 1991, commencing at 10:00 a.m.

PRESENT:

O'Sullivan, Graev & Karabell
(by Henry B. Gutman, Esq.)
30 Rockefeller Plaza, New York, NY 10112,
for the Plaintiff.

Fenwick & West
(by Edmond C. Gregorian, Esq., and
Stuart Meyer, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[2-131] Q. Simple.

A. That doesn't mean anything to me. In the network example—once it's specified and standardized, you might describe it in one page, and it might look very simple; but it might have taken years to get there. So I'm sorry. I can't respond in that way.

*Q. Within applications software, is there any generalization that can be made concerning the percentage of time—the percentage of total design time that would be spent on user interface design?

MR. GUTMAN: Object to the form.

A. Are you assuming in the words "applications software" that there will be a human interacting with a presentation on a screen?

Q. Yes.

MR. GUTMAN: And you are speaking in terms of time, not effort measured qualitatively?

MR. GREGORIAN: Just quantitatively.

A. Okay. Could I have the question read back.

*(Question read)

A. I find it hard to make a generalization.

* * *

[2-183] (Document marked as Exhibit 11 for identification)

Q. Doctor, I'm showing you an excerpt from Brooks' book, *The Mythical Man-Month, Essays on Software Engineering*, which I've marked as Exhibit 11. Take a look at page 20, sir.

A. (Witness reviews document) Yes.

Q. Do you see the paragraph that begins, quote, For some years?

[2-184] A. Yes.

Q. Does that paragraph refresh your recollection in any way as to whether Mr. Brooks' work is the source of your rule of thumb that's described in your affidavit?

MR. GUTMAN: Object to the form. I don't believe you've asked him whether it was the source. You've asked him whether it was consistent.

MR. GREGORIAN: I am now asking him whether it's the source.

A. All right. So you are not asking me whether it refreshes my memory or helps me recollect; you are asking me, was it the source?

Q. Yes, sir.

A. I can't say it was the source. I'm delighted to see it. I haven't looked at this book for many years, and I'm delighted to see that I am consistent with it. Whether it was a source or not, I don't know.

* * *

[2-233] Q. Now, are you aware, Doctor, of any rules or criteria for user interface design?

MR. GUTMAN: Object to the form. What do you mean, rules or criteria?

MR. GREGORIAN: Design rules, design principles, design criteria that are applied by people, practitioners, in designing user interfaces.

A. Okay. Now that you have expanded the question, could I have—would you restate it as a single question.

Q. Are you aware of any design rules, principles, or criteria that are applied by practitioners in the field of user interface design?

MR. GUTMAN: I object to the form.

A. There are some common sense rules about ease of use and so on. I don't know of any specific rules that are applied.

Q. What do you mean by "common sense rules about ease of use"?

A. Such as it—the product ought to be easy to use.

Q. Anything more specific than that?

A. No.

[2-234] MR. GUTMAN: I object to the form.

Q. How about with regard to menu structure, organization, and design—strike that. How about menu design—are there any rules, principles, or criteria for good menu design that you are aware of?

MR. GUTMAN: Object to the form.

A. I would expect that there would be such rules for the people who are directly involved in those activities. I am not aware of them.

Q. Are you aware of any literature that discusses rules or design criteria for user interface design?

MR. GUTMAN: I'll object to the form of the question. He identified yesterday, I believe, literature he was aware of or some literature he was aware of concerning user interface design.

MR. GREGORIAN: Generally, that's correct.

MR. GUTMAN: But that's what makes this question confusing. Are you asking whether he remembers whether those books include a reference to rules?

MR. GREGORIAN: No.

Q. Other than what you mentioned yesterday, are you aware of any literature on user interface **[2-235]** design principles and criteria?

MR. GUTMAN: I object to the form.

A. No. I told you yesterday what I remembered.

Q. Calling your attention to Paragraph 51 of your declaration, it states, quote, Within software specification and design, expression of the user interface is probably the most critical task the application developer faces and that in which the most significant creativity—creative activity and effort is expended, closed quote.

Now, in that statement you refer to application developer. Are you limiting that to application software development?

MR. GUTMAN: I object to the form. You can answer.

A. I don't think the intent was to limit it. I believe it was put in there because that was the context of the discussion.

Q. So you would believe that the statement has more general applicability than just applications software; is that correct?

MR. GUTMAN: I object to the form.

A. I think it has more general applicability

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Volume III
Pages 3-1 to 3-257
Exhibits 12-13

CIVIL ACTION
NO. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—VS.—

BORLAND INTERNATIONAL, INC.,

Defendant.

SPECIALLY CONFIDENTIAL

CONTINUED CONFIDENTIAL DEPOSITION OF BERNARD A. GALLER, a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Anne H. Bohan, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Friday, August 30, 1991, commencing at 8:55 a.m.

PRESENT:

O'Sullivan, Graev & Karabell
(by Henry B. Gutman, Esq.)
30 Rockefeller Plaza, New York, NY 10112,
for the Plaintiff.

Fenwick & West
(by Edmond C. Gregorian, Esq., and
Stuart P. Meyer, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[3-80] Q. By the person who developed the product. In other words, is it typical to test a user interface design at some point prior to product release with human beings?

* * *

[3-81] Q. I meant as a discrete test.
A. Specifically for that purpose?
Q. Yes.
MR. GUTMAN: Object to the form.
MR. GUTMAN: A. You did say by a person who is a developer, I think, and of course in an organization the testing may very well not be done by a person.
Q. All right, the company or whatever that is involved in the product.
A. Yes. I believe it is typical to ask people to interact with the product so as to ascertain the effectiveness of the user interface.
Q. So that would be testing where people are invited to use the product and then comment in some way on the user interface?

* * *

A. That depends on the experimental design. I don't know how it might be done, that might be one way.

Q. Do you know what is done or the range of [3-82] things that are done in that type of testing?

* * *

A. No.
Q. Do you know what manner or means are used to obtain feedback from people who—
A. Well, in answer to your previous question—

* * *

A. —when I said I did not know, I had in mind within industrial companies; I mean, I have been involved in situations where I have done testing on software that I have done, and I know what I did to test it, but I answered your question in the context of typical industrial internal processing.

* * *

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

PAGES 1-179
EXHIBITS 1-5

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION, *Plaintiff*,

v.

BORLAND INTERNATIONAL, INC., *Defendant*.

Rule 30(b)(6) Deposition of
Lotus Development Corporation by its designee
Ezra Gottheil, Ph.D.
September 22, 1992
Boston, Massachusetts

PRESENT:

Kerry L. Konrad, Esq.
O'Sullivan Graev & Karabell
30 Rockefeller Plaza
New York, New York 10112
for Lotus Development Corporation

Peter E. Gelhaar, Esq.
Donnelly Conroy & Gelhaar
176 Federal Street
Boston, Massachusetts 02110

—and—

Peter N. Detkin, Esq.
Wilson Sonsini Goodrich & Rosati
Two Palo Alto Square
Palo Alto, California 94306
for Borland International, Inc.

Ezra Gottheil, Ph. D.

* * *

[3] Q. Mr. Gottheil, could you state your full name for the record, please?
A. Ezra Lee Gottheil.
Q. Do you reside in or around Boston?
A. Winchester. It's about eight miles north.
Q. Could you give me your address, please.
A. 4 LaGrange Street.
Q. And you are employed by Lotus?
A. Yes, I am.
Q. What is your job title?
A. I'm a senior product design manager.
Q. And in that job title what do you do, basically?
A. My job is to coordinate the selection and design of features in our products, what features go in, how they're implemented, what the user interface looks like.
Q. How long have you had that job?
A. In its current position I've had it since February of '89 on different products, and then I was previously a designer for a period of time back in '85.

* * *

[4] Q. When did you first start at Lotus?
A. July of 1981.

* * *

Q. Can you somewhat stream of consciousness go from the time you first started with Lotus, with the company that you've just corrected me and was not Lotus, but could you tell me what your jobs were, when you had them, and the job descriptions?

[5] MR. KONRAD: I think that by "stream of consciousness" Mr. Gelhaar is asking you to give some sort of narrative. Typically that's not what we do, but I think he is just trying to save time. So if you can work through that chronologically, go ahead. But if you have problems doing that, Mr. Gelhaar can break it down, if necessary.

A. I'll give it a try. By the way, I was working part-time three days a week for the first year and a half and then four days a week for about another year until I became full-time.

Q. Let me break it down right now then. What were you doing when you were not working with Mr. Kapor?

A. I was working as a psychologist at Human Resources Institute in Lowell, Massachusetts, and I was working on my dissertation.

Q. So essentially you would work on that two days and you'd work with Mr. Kapor three days?

A. Right.

Q. Sorry for interrupting you. Why don't you continue, please.

A. Quite okay.

So my first job, which had no title [6] because it was a company of four people, was designer and programmer of a project that I proposed and set forth to implement.

Q. What product was that?

A. I called it IMP, the idea manipulator. That was one of its many names. It never saw the light of day.

After about a year, somewhere in early summer of '82, I requested that I work on 1-2-3, what was to become 1-2-3. It was then code-named Trio, I believe.

Q. Incidentally, was IMP a spreadsheet product?

A. No, it was a hypertext product.

Q. What is a hypertext product?

A. Hypertext is a system by which text is linked to other text and you can move from one piece of text to another along these links.

Q. Why did IMP not see the light of day, as you say?

A. It was at my suggestion that I stop working on IMP and start working on 1-2-3. There were two reasons for that. First of all, it became apparent to me that the company was and needed to be [7] focused on what was to become 1-2-3, and the other was that there were people around with far superior programming skills to my own and I was better off doing something else.

Q. I believe you told me we are at early summer of 1982 when you began working on what you referred to as the product Trio, which was the code name for 1-2-3.

A. Mm-hmm. At times it had other code names, but that probably is what it was at that point.

* * *

[127] Q. What was your involvement, if any, in the development of the menu command hierarchy of version 1.0?

A. I made some suggestions. As a documenter and a member of the team, I was free to make suggestions about how I thought things might be made [128] better, both verbally and in writing.

Q. Did you have any decision-making capacity, in other words final decision-making capacity, concerning the development of the menu command hierarchy of version 1.0?

A. No.

Q. Are you familiar with the process by which the version 1.0 1-2-3 command hierarchy was decided?

A. I am aware of some of the considerations that went into it and I believe I understand what the various people did, but I was not present when these decisions were made.

Q. Could you tell me the considerations that you're familiar with?

A. I'm familiar with the wish to group commands logically. I am familiar with the wish to make commonly used commands be near the top of the menu, to put them at the left edge of any menu where they are—The more common, the farther toward the left they should be; that the words that are used should as clearly as possible communicate what the command will do; that the first letters of all the commands at any given menu be different; and that ideally no more than seven commands be in any menu.

[129] Q. Any others?

A. With respect to the choice of words, not only should they be clear but they should be as untechnical as possible.

Q. Any others?

A. An aesthetic judgment comes into play; simply a sense of words being right, feeling right, sounding right.

Q. Can you elaborate on that one for me, please? What do you mean by that, the aesthetic considerations of a word?

A. I don't think I can go much beyond it. When one chooses words, one chooses them for every purpose for their sound, for their pronunciation, for their resonances and reminders. One might, for instance, use the word "kill" to describe a removal or deletion, but it is an ugly word with negative associations.

Q. Would an example of an aesthetic quality of a word or in terms of choosing command words be that words should or should not rhyme?

A. Possibly. I've never heard of that being an issue, but it actually makes some sense.

Q. Would another example be that command **[130]** words not be alliterative next to each other?

A. In what sense?

Q. That they sound too similar.

A. Again, a worthy consideration that might come up under review or inspection but not one I've ever heard explicitly announced.

* * *

[131] Q. Was screen size a functional constraint with respect to menu commands?

A. I'm not sure what you mean by functional constraint. There was a limited screen size that was set by the machine we were running on, which had 80 characters by 25 lines.

Q. And the fact that the first line only had 80 characters or each line, actually, only had 80 characters, how did that constrain the choice of menu commands?

A. Clearly, no menu could be longer than 80 characters. In fact, because we wanted to keep them shorter than seven commands, we had other constraints that were more powerful.

* * *

[132] Q. Incidentally, was the menu command hierarchy reordered shortly before the release of version 1.0?

A. I do not know at what phases in the development of the product reorderings occurred. I do know that periodically things got changed and we would work with a given ordering and hierarchy for a while until certain objections surfaced, and then change them, and I don't remember any particular last-minute or toward-the-end kind of reorganization. One of the constraints on how much one can do that is the documentation must be finished before the software, so that it can be printed. So at a certain point everything that's described in documentation is set.

Q. Do you recall how many times the order of the commands was changed during that process in the **[133]** summer of '82?

A. No. I don't

Q. If you had to guess, what would be your best guess?

A. My best guess is that there were very few complete rearrangements, but there were many modifications to one or two commands at a time. But I really have no idea of the number. Small changes were being made relatively frequently. Large changes were less frequent.

Q. So, in other words, more tinkering than large wholesale changes of the menu commands?

A. That would be my recollection.

Q. You testified earlier that it was a wish to group the commands in a logical fashion. Is that fair to say?

A. Yes.

Q. Could you explain what you mean by that?

A. In a hierarchical menu what exactly are the commands is not apparent when the user is at the top level. There are a series of operations that can be performed, but you don't have an exhaustive list before you. Therefore, what you try to do is put them in a category where the user is most likely to **[134]** find them if he or she looks for them there.

So what you want is a clustering of commands that either do similar things or operate on similar objects so that they can be described by one word, and in most cases the user will be able to infer, if they want one of those commands, to look under that word.

Q. Do you recall any specific discussions with anybody during the summer of 1982 and up until the release of 1.0 where this desire to have commands flow logically was discussed?

A. I remember only one discussion about specific menu issues and it had to do with putting move and copy underneath the range command. That would have given us more room at the top level and put them in a place they would be logical. But the constraint that said they should be as close to the top as possible conflicted with the constraint that said they should be grouped together.

* * *

[135] Q. Can you give me a concrete example of the menu commands being grouped in a logical fashion so I can focus in on your point here?

A. Among the most obvious is file, where retrieve and save and combine and extract and directory are all put in one place. It makes complete sense to the user that anything having to do with a file is done there.

Q. And as a person who was developing the menu commands, you wouldn't want to do anything that didn't make sense to the user. Correct?

A. Right.

Q. So from that standpoint it is fair to say that the person or persons who developed the menu commands were constrained to do so in a logical **[136]** manner. Correct?

A. There were constraints like that, but there was enormous freedom of choice as demonstrated by the constant discussion and changing and also demonstrated by products that do the same thing but yet have different menu structures.

Q. It is fair to say, though, that there was no freedom of choice to be illogical?

A. There was no freedom of choice to be completely—Well, there was no freedom of choice to be completely illogical if a logical choice were available. In some cases no logical choices were available and you were making the best of a bad situation.

Q. And making the best of a bad situation was always picking the most logical of potentially illogical choices. Correct?

A. With the other constraints all in operation as well.

Q. You mentioned that you wished to make commonly used commands appear at the top.

A. Yes.

Q. Can you explain what you mean by that for me?

[137] A. In a menu there is the first level of commands, those that are chosen after the command prefix, in our case a slash key. We call that the top level. We want the most frequent things to be there.

Q. How did you arrive at that conclusion, that that should be the way it should be ordered?

A. I did not do this. I believe the way it was done was the introspective method. That is to say, the people who were designing the product used the product a great deal and noticed what they did. We also received some feedback from talking to users outside the company and we would ask—I presume we would ask them what they did most often.

* * *

[143] Q. Dr. Gottheil, you mentioned in connection with the consideration we were most recently speaking about concerning most commonly used words being placed at the top of the line that more familiar commands should be placed at the left edge of that line. Did I get that correct?

A. I think it's more commonly used commands that should be placed at the left edge. The assumption is that the eye scans from left to right and the sooner one finds what one is looking for, the happier one is.

Q. Do you recall hearing any discussions among the Trio team concerning that consideration?

A. No, not specifically.

Q. Generally?

A. I don't recall specific discussions. It is my impression that that was one of the guidelines, and I don't remember exactly how I gained that impression, but I'm fairly sure it was a common consideration.

Q. Would one example of that constraint be the word "worksheet" appearing in the far left?

* * *

[145] A. Frankly, I don't know why "worksheet" appears in the far left. My guess would be that it is used less often than some other commands.

Q. What command would you have rather seen appear on the first command to the far left?

A. It's a guess that it is less frequently used. There might be other reasons to leave it there. For instance, it may be that you want the user to see the list of choices, which in the case of **[146]** worksheet are a kind of odd assortment of things that are available under that command. When the cursor first starts up, that command is highlighted and that prompt is displayed. But in terms purely of frequency of use, my guess is either copy, move, or range would be used more often.

Q. Can you give me an example, if they're different than the ones you just gave me, of having a more familiar command appear on the far left?

MR. KONRAD: Objection. Again you said "more familiar." Do you mean "more commonly used"?

* * *

[147] Q. Did you use the word "familiar" or "more commonly used"?

A. If at any point I used the word "familiar" in this context, I meant "more commonly used." If indeed commands were familiar, we would not have to go to pains to make them more obvious.

Q. Then would you give me an example of a more commonly used term which it was decided should appear on the far left-hand side of the screen?

[148] A. I don't know.

Q. And the reason that you don't know is because you were not involved in that portion of the process of the development of the command hierarchy. Is that correct?

A. There are two reasons. The first is that I was not involved. Secondly, that there are a number of considerations as to what goes where, and that unidimensionally singling out frequency of usage or shallowness in the menu or any of another bunch would suggest that one can generate an interface from rules. One cannot.

Q. Why not?

A. I misspoke myself. One can generate an interface from rules. It would be a bad interface. One cannot generate a good interface from rules. The point of this is that it is an artistic creation of one or a group of dedicated professionals to make a good, usable interface.

Q. And those principles of expression, as you say, are certainly constrained by certain logical ways of performing this task. Correct?

* * *

[149] A. Are there things one cannot do in designing an interface, or one should not do? Yes. One shouldn't use a word that is completely irrelevant to the command. There are a thousand other things that one should not do. The space of bad choices for an interface is vast. The space for good choices is quite large as well.

Q. What you just said, the thousand other things that one should not do, it is fair to say, is it not, that those are constraints on a developer of a menu command hierarchy. Is that correct?

* * *

[151] Q. I will agree with Mr. Konrad that if you cannot answer that with a yes or no, then you need not. If you can answer it yes or no, I would like you to do so.

A. I have become quite sensitive to what you mean by a constraint. Let me say what I believe to be the case.

There are many choices one can make which would make the interface bad: word choices.

* * *

[152] Q. Dr. Gottheil, another rule or guideline which I will— Is it fair for me to say rule or guideline in the conversation we have been having as opposed to constraint? I mean, would that speed things along?

[153] A. Guideline would be preferable to rule.

Q. Fine. Concerning the guidelines that you initially discussed, I believe you said that the words should communicate intelligently what the command will do or something to that effect. Is that fair to say?

A. I'm sure I didn't use the word "intelligently." The meaning I was trying to communicate is that the new user would have the best possible chance of determining what the command will do from its name.

Q. Tell me why "global" is intuitive then.

A. In fact, I don't believe "global" is intuitive. It may very well be the least unintuitive of the choices, given the other

guidelines and your attempts to balance everything. "Global" has meaning. It means affects everything. But in many cases it would not become apparent to users what one might affect having to do with everything, certainly not from the word.

Q. Do you recall any specific discussions concerning the choice of the term "global"?

A. No, I don't.

Q. Do you recall any discussions concerning **[154]** the guideline of choosing words so that they will communicate to the initial users what the command will do?

* * *

A. This would be a case where we change the name of a command, and I am sure we did, but I don't remember any specifics. I can say that there are differences, for instance, between the wording in Symphony and 1-2-3 where my guess is we thought we had an opportunity to use a more clear word, "erase" versus "delete." Then there is "exit" versus "quit." In Symphony, it is exit, which is less clear, but probably the Q is taken for something else.

Q. What was the Q word taken for?

A. In Symphony, it was probably "query." So we needed a word other than one that began with a Q that meant stop the program and leave, so we used "exit," which is less clear than "quit."

Q. I believe you referred to a concept that you called shallowness in the menu.

[155] A. Mm-hmm.

Q. Could you tell me what you mean by shallowness in the menu?

A. Yes. Requiring the fewest keystrokes; being nearest the top of the menu hierarchy.

Q. And can you expand on that? What does that mean to the user?

A. It means two things. One is, the first is that you need to type fewer keystrokes to get to it, and that is useful when you use the same thing over and over again. It also means that if

you are looking around for it, you're more likely to find it sooner rather than later.

Q. So I take it that there is a guideline which essentially states that the author of a menu command hierarchy should permit the user to get to the end of the tree as fast as possible? Is that correct?

* * *

A. You can't get to the end of the tree all the time as soon as possible. Some things have to require more keystrokes, which will take you longer, and others shorter. In general, the guideline is: **[156]** The more often you do something, the sooner you want to have the user reach that end of the tree.

Q. Can you give me an example where that guideline was put into practice with respect to version 1.0?

A. Yes. This is one of the rare cases where I remember one. In the case of the move command, it's slash move, and it does it; and copy is the same way. I, for instance, was advocating making it slash range move, two keystrokes, with the negative effect of requiring an additional keystroke and requiring the user to look for it but with the positive effect of consolidating our top-level menu.

* * *

[158] Q. I believe you testified, Dr. Gottheil, that another guideline would be that the first letters would have to be different. Is that correct?

A. Under the design guidelines that we used for 1-2-3, yes. Now, that is a guideline that has been frequently violated by other products at other times, but we were working from that assumption.

Q. How can that rule be violated?

A. It can be violated by highlighting in some way other than the first letter in a command and make that the selection key. That is the convention, for instance, under Windows. The drawback is that now **[159]** the user has two things to remember: the name of the command and the activating letter.

So it is clearly preferable to have the first letter be the activating letter.

Q. Was an example of the guideline in practice of not having first letters be the same in command terms the words "erase" and "extract"?

A. That's the only violation I can remember in the original 1-2-3.

Q. Can you explain to me that potential violation and how it was solved?

A. In that case it was decided that for "extract" we would spell it without the initial E and therefore the first letter would be X.

* * *

Q. I believe you also testified, Dr. Gottheil, that there was a guideline that no more than seven commands should appear in any menu line. Is that correct?

[160] A. That is correct.

Q. What is the basis of that guideline?

A. There is an article by Herbert Simon called The Magic Number Five Plus Or Minus Two in cognitive psychology dates back, I think, to the early '60s. I've recently reread the article. It doesn't quite say what we thought it said. Nevertheless, it has been a useful guideline because the purpose is to keep the number of choices available to the user at any one time to a small enough number so that they can find their particular choice quickly.

Q. Was that guideline ever violated with respect to 1.0?

A. Frequently.

Q. Can you give me an example?

A. The top-level menu, the worksheet menu, the format menu, and probably many others.

Q. Concerning the worksheet menu, why was it violated, if you know?

A. I don't know specifically with respect to the worksheet menu or any other. The reason it got violated is because it was felt that it was better to put a few more commands on those

particular levels **[161]** than any of the other choices, to make the menus deeper or to move them to other menus.

BY MR. KONRAD:

Q. Just for clarification, when you say worksheet menu there and format menu there, you mean the submenus branching from those two commands in the top-level menu?

A. In the case of worksheet, from the top level; the things under worksheet. In the case of format, it is under range format and those are the choices under there. There are more than seven there.

MR. KONRAD: Thank you.

BY MR. GELHAAR:

Q. And with respect to the violation that you just mentioned where more than seven commands appeared on the menu line, did that guideline give way to another guideline?

A. It gave way to a general sense that the other guidelines singly or together—I don't know which in any one case—overrode that one. Always a judgment call, always an aesthetic judgment, a judgment of what seems and feels right, what falls readily to hand when you use it.

[162] Q. Another guideline you mentioned was that words should be as untechnical as possible. Did I get that correctly?

A. Yes.

Q. Can you give me an example in the context of the creation of 1.0 menu command hierarchy where that guideline was put into effect?

A. No, I can't.

Q. Do you recall any discussions with anyone during the creation of 1.0 where that guideline was discussed?

A. I cannot recall specific discussions and therefore I cannot recall the content. I recall that being an argument used in favor or against choices, but I don't remember what the choices were.

Q. So you recall that just generally. Is that correct?

A. Yes, yes. And the kind of exchange would be "Only a tech weenie would understand what that means."

Q. And lastly from my list of guidelines that I believe you testified about was what you referred to as aesthetic considerations of words and their order. Is that correct?

[163] A. Yes.

Q. I believe you testified about the aesthetic considerations of the word, but could you tell me what you mean by the aesthetic considerations of the order of the words?

A. I have no specific example. I just have a sense that in all cases you step back and you say, for instance, could a pairing of words be construed to have a meaning together? You just look for those things or they get pointed out to you during the testing of the product by the team saying, no, you can't put those words together. Let's separate them. They're too confusing. They mean something together that we don't want to imply, whatever like that.

* * *

[164] Q. I guess what I'm getting at, Doctor, is: Are there any other guidelines that you have employed at your work at Lotus that are in addition to those which you've just mentioned?

A. Probably, but they don't come to mind. As I say, it is like writing; it is sitting down and trying to communicate. And the primary thing one does is try to adopt the point of view of the new user or the user new to that part of the product, and just say what would I think. So you don't go through a series of rules and generate an interface. You write something down, look at it and say, does this tell me what I need to know? Would I understand what I'm about to do? Could I get myself in trouble? That sort of thing.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF FRANK A. INGARI, JR., a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure before Carol H. Kusinitz, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Tuesday, August 6, 1991, commencing at 10:05 a.m.

PRESENT:

O'Sullivan Graev & Karabell (by Henry B. Gutman, Esq.) 30 Rockefeller Plaza, New York, NY 10112, for the Plaintiff.

Fenwick & West (by Mitchell Zimmerman, Esq., and Lisa A. Olson, Esq.) Two Palo Alto Square, Suite 800, Palo Alto, CA 94306, for the Defendant.

* * *

[101]

* * *

Q. You testified that Lotus would not want to **[102]** change its command structure because "we consider it to be one of the most precious assets we have." Do you see that testiomony?

A. I do.

Q. By "command structure" there you were referring to the names of the commands in 1-2-3's menus and the way the commands were arranged in the menus; is that correct?

MR. GUTMAN: Object to the form.

A. We mean the particular commands, the word choices of the commands, the first letters of the commands, the arrangement of the commands in a particular moving-cursor menu, with appropriate associated submenus, that complete complex of menu choices and their arrangement.

Q. Okay. Why did you consider that to be one of Lotus's most precious assets?

MR. GUTMAN: Object to the form. Why did he at the time of his testimony? Is that the time frame you were addressing?

Q. At the time you made the statement, "We consider it to be one of the most precious assets we have," why did you consider the command structure—

MR. GUTMAN: As of February of '89.

[103] A. Lotus 1-2-3 is sort of like the Beatles. It is a phenomenon. There are millions of users. The collective usage of this product by millions of people amounts to a sort of collective wisdom of how to achieve functional benefits through a specific expression of a software product. That knowledge base, fingertip knowledge as it's called, was a tremendous advantage to us, because we believed and I still believe, as a private individual, that people are more willing to stay with a software program which they understand and with which they are familiar at a physical tactile level.

In other words, hundreds of thousands of people type 1-2-3 as you and I might type on a Qwerty keyboard. They know those sequences of commands in their fingertips, if you will. That was an extremely strong incentive for users to stay with Lotus, and to upgrade to future versions of our products, and perhaps to purchase services from our company.

Secondly, because we believed that the particular design work that had been done by Jonathan Sacks and Mitch Kapor was particularly brilliant, that it was a brilliant and economical **[104]** expression of a way to work with numbers in rows and columns. It wasn't the only way, but it was very fast, very compact, very elegant.

Q. Is it fair to say that you did not consider the command structure to be one of Lotus's most precious assets because the arrangement of commands into menus and submenus was more logical than the arrangement of commands in menus and submenus of other spreadsheets?

MR. GUTMAN: Could I hear that back, please.

(Question read)

MR. GUTMAN: I'll object to the form of the question.

A. I don't recall ever having a conscious thought about its logical quality versus someone else's expression of a way to achieve the same result. My beliefs were founded on 1-2-3 itself. You have to remember this product dominated in terms of installed base, and we were more focused on our customers and their feelings than we were on whether it was more logical than another product. We felt that this asset was uniquely strong. Even if someone had brought out a different expression that **[105]** was more logical, we felt very strong with what we had established.

Q. I mean, even if there were another way of organizing commands into menus that was more logical, that what made this a precious asset in your view was that millions of people were used to it?

MR. GUTMAN: Object to the form.

A. No, as I said before, there were multiple reasons why it was a precious asset. One is it had been so successful, which in many ways was a proof of it, and two was that it was extremely elegant, compact and efficient. The word "logical" is your word. It is not one that would in fact occur to me. My word would probably be "intuitive" as opposed to "logical."

But those are words. Those are opinions. It is not only because millions of people used it. It was not because it was a standard so much as that it was—there were reasons why it was a standard, which was it was damn good, which is how it became a standard.

* * *

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF BECKY D'ADDEA JONES

Thursday, September 5, 1991

Reported by: Darleen J. Henry, CSR #3599

---

* * *

[27] Q. Well, I'm asking for your view about the marketing to this kind of account, and I'm saying—what I'm asking is, do you recall whether you had a view back at the time before Quattro came out that you needed to have the 1-2-3 menu trees in Quattro in order to reach those customers?

A. My view was that we needed to protect what they ha[d]. That meant I wanted to make sure that their files could be read and that their macros could be used. How technically we got there, that was not my concern.

Q. Trying again to focus on the time just before Quattro came out, and I realize it's difficult and artificial to some extent at this point, but trying to do that, can you recall anyone else at Borland expressing the view that we have to have this 1-2-3 compatibility in order to reach those corporate accounts?

A. I don't recall.

* * *

[116] Q. On the cover memo in the last paragraph you refer to several individuals. There are a couple of new names there. One is Roy Kayn, K-A-Y-N. Who is he?

A. Sales.

Q. And there's a reference to a Mark Andrews; who is he?

A. He was in charge of training.

Q. Training for whom?

A. He did internal training on Borland products as well as demo disks.

Q. What department was he in, do you recall?

A. He was part of the sales organization.

Q. Do you recall who was head of sales at this point in time, mid '88?

A. Doug Antone.

Q. And he still is today?

A. Yes, he is.

Q. Excuse me. If you'd look at the page that's stamped 519315, the top section on "Compatibility," at the end of it there's the sentence that appears in the later version about habitual keystrokes.

[117] There's a sentence on a special section in the documentation that's similar to a sentence in the later version. The last sentence, though, reads, "Therefore, no retraining, no relearning is necessary." Do you see that?

A. Yes, I do.

Q. Did you write that?

A. Yes.

Q. Were you referring there to the point that 1-2-3 users wouldn't have to be retrained or learn a new menu tree to work with Quattro Pro because of the 1-2-3 menu tree?

A. Yes.

Q. Now, you can compare them if you want. I think counsel has the previous one. But that one sentence—there may be some wording changes in the other sentences—but that one sentence has not a draft date. Any reason why?

A. These things get tweaked all the time.

Q. I understand, and I could not possibly myself explain all the changes that I have ever made in editing documents. I'm just wondering if you can recall any reason for that one being dropped?

A. No. Like I said, these get tweaked.

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Case No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF PHILIPPE KAHN

Wednesday, May 22, 1991

VOLUME I

Reported by: Deborah A. Fryer, CSR #5191

---

* * *

[57] Q. You testified, actually I don't believe it [58] was even in response to a question, that the reason you put the 1-2-3 compatible menu trees in Quattro and Quattro Pro was for purposes of macro compatibility?

A. That's correct.

Q. Was another purpose to take advantage of the fact that there are many users out there who know how to use 1-2-3 and are familiar with the sequences of keystrokes necessary to accomplish certain tasks using 1-2-3?

A. No.

Q. So you were not trying to take advantage of that learning curve or experience that 1-2-3 users have?

A. No.

Q. That was no factor at all in the decision?

A. No.

Q. And your answer covers both Quattro and Quattro Pro?

A. That's correct.

Q. Is that still not a factor today?

A. That's correct.

* * *

[65] A. My recollection at the time where we made the decision that we needed to have the 1-2-3 compatible menu tree in order to run macros was because studies that were published by newspapers, magazines at the time, if I remember Adam Bosworth and I looked at that and I think what we concluded was that, if I remember well, over 50 percent, I think, I think it was over 50 percent of the people, maybe 60 percent, I don't recall exactly, of the people using the spreadsheet 1-2-3 actually made use of worksheets that included some form of a macro.

Q. And did that information play any role in your decision to incorporate the 1-2-3 menu trees?

A. It was important, yes.

Q. Would you say that was the most important factor in the decision?

A. The information you're talking about for running the Lotus macros?

Q. Yes, that information.

A. That information was important together with some other information that I'll describe in a second, because it

showed us that for this product to **[66]** be marketable we needed to be able to execute macros.

When we brought the product beta testing, I recall that beta testers would tell us that we couldn't run some of their macros because the beta test versions of Quattro Pro right before it shipped did not include a 1-2-3 menu tree. Actually we didn't have one and we couldn't execute the macros.

And those two facts really against our will, I remember discussing this with Adam, those two facts really showed us that macro compatibility was the key to being able to have a product that could have any chance in the marketplace.

Therefore, almost at ship time Adam and I basically decided that we had to build, using the Menu Builder, a 123.RSC file, a resource file, that would allow as many as possible of the 1-2-3 macros to execute properly.

* * *

[95] Q. Now, back in the context of your conversation with Bosworth, you referred to a belief that, and again, I'm just trying to put you back in context, your testimony is in the record and that controls, but you said that for this product, referring to Quattro, to be marketable, it needed to be able to execute macros.

Were the macros you were referring to there, sir, 1-2-3 macros, or was it sufficient for Quattro to be marketable for it to be able to execute Quattro macros?

A. I'm referring to 1-2-3 macros.

[96] Q. Is it still your belief that in order for Quattro to be marketable or Quattro Pro to be marketable it needs to execute 1-2-3 macros?

A. I don't know.

Q. You don't have any view on that subject today?

A. What do you mean by "view"?

Q. Belief, opinion, notion?

A. It's a difficult subject, complex.

Q. That isn't the question, sir, do you have a view or opinion or belief on that subject?

A. No, that's my answer, it's a difficult subject and I don't usually express beliefs that cannot be, you know, at least reasonably understood.

Q. I'll try it a third time: Do you have a belief on the subject of whether Quattro or Quattro Pro in order to be marketable still needs to be able to execute 1-2-3 macros?

A. I don't know.

Q. At the time of the conversation with Bosworth, did you discuss any ways to permit Quattro to run, or excuse me, to execute 1-2-3 macros without incorporating the 1-2-3 menu tree?

A. Yes.

* * *

[99] MR. GUTMAN:

Q. Yes, and my question to you was did you discuss with Bosworth whether it was possible to achieve the ability in Quattro to execute 1-2-3 macros by means other than the incorporation of the 1-2-3 menu tree, your answer was "yes," and I then asked you to tell me about that conversation, and I will now repeat that question: Please tell me whatever it is you recall about that conversation.

A. Sure. We discussed the fact that Lotus macros can be interactive, in other words can display commands on the screen, and therefore it became obvious that in order to execute 1-2-3 macros you had to build a menu tree similar to the one that is encapsulated in the 123.RSC file.

* * *

[100] A. We also at that time realized that macro translation was not a viable way to do things because Lotus macros can be self modifying. In other words, because a macro can modify itself, if you use the stand-alone translator, then of course the translated version of that macro couldn't modify itself.

The analogy I can use here is the analogy of French and English. Once I've translated French into English, I can't interactively keep on speaking French, it's English, and the point that we made, just as you yourself, Mr. Gutman, couldn't understand my definitions when I wrote them in French on my piece of paper, I'm sure you'll agree that, well, the same thing is true for translating macros. So we realized that self-modifying macros ruled out the possibility of using translators.

On top of that one important feature that [101] Quattro Pro, or Quattro, sorry, it's not Quattro Pro, it's Quattro, was doing by design that no spreadsheet did at the time, to the best of our knowledge, was macro debugging.

And when it came down to macro debugging it became obvious that macro translation didn't work either. I think that Quattro Pro, or Quattro, sorry, was the first product ever, or first spreadsheet, I think ever to do debugging. I think that Lotus 2.3 now, several years later, has added that feature.

So as far as I recall, those were the three kind of points that we discussed.

* * *

[103] Q. Did he ever suggest to you other alternatives that he had considered?

A. That he'd considered?

Q. Yes.

A. What do you mean by "considered"? Looked at?

Q. Looked at, contemplated, analyzed; pick your favorite synonym.

A. I just gave you the reasons why others couldn't be really seriously looked at, I described extensively why, logically, any person who's technical enough to know that understands that translation doesn't work, actually I did more than answer your question.

If you're a logical person you do understand at that point that we looked at many ways to do things and the only reasonable way, for the reasons I just expressed, was—that you could execute in a reasonable way macros was to actually

incorporate a file such as 123.RSC with the shipping version of Quattro.

* * *

[118] personally, I mean I have done, I have done personally thinking for many years about language translators.

What I mean by "language translators" is compilers, debuggers and all that, that's how Borland started, and in that respect give a lot of thought to these kinds of issues, not specifically to the Lotus case, but they apply to areas such as these.

So if you're asking whether I have done and others of the company have done this kind of thinking, it's part of what we do when we design compilers, when we design interpreters, when we design debuggers, and in that respect, you know, the answer is as I gave it to you before, because you have self-modifying code.

If I'm going to use the terminology that's used usually, which is what you call self-modifying macros and interactive macros, there's not many ways to do it, and the rest is pure speculation.

MR. GUTMAN:

Q. Have you made any effort today to determine the extent to which people actually make use of self-modifying or interactive macros or macro debugging? And by "you" I mean you personally or, to your knowledge, Borland?

A. There are clearly on the product, I mean on **[119]** the market, commercially-available templates that you can buy from third-party companies that actually use these, I'm sure there are, and you know, interactive macros are part of every menuing system built with macros in a spreadsheet, as such.

So I mean it's pretty obvious that it's a feature that's important to users, so you know, it's a well-known fact in the industry.

Q. So you think there's an industry consensus that these are important issues for users?

A. I don't know what you mean by "industry consensus."

Q. Well, you say it's a well-known fact in the industry. My question is is this something upon which all right-thinking people in the industry agree, that is that these three issues, self-modifying macros, interactive macros, and macro debugging, are important to users?

* * *

[120] A. So, I'm just answering your question, no one ever agrees completely in the industry, you'll always find people who disagree with opinions, that's why this industry is so diverse.

So the answer to this question, to the question as you put it, is can't answer that. What I'm saying is it's pretty clear to people who use spreadsheets and macros that interactive macros are important, and for some interactive macros are important, for some self-modifying macros are important, to others macro debugging is important.

* * *

[146] it's going to hurt us"?

A. I can't recall, it's possible, I answered that question, I can't answer it three times. I answered it, it's possible, I can't recall precisely.

I talk to a lot of people every day about a lot of subjects, and believe it or not this issue is not one that I spend my life on.

Q. Well, was it an important issue during the week after we sued you?

A. Certainly was an important issue.

Q. Was it an issue that you focused substantive attention on during that period of time?

A. What do you mean by "substantive"?

Q. Material attention?

A. Listen, I spent a few hours on it, that's all, I have a company to run and lawsuits don't help me run my company, so I don't—I delegate that kind of thing, that's why I have counsel. I trust that they will come up with the best advice.

Q. But your outside counsel didn't participate in those discussions, did they?

A. I don't know. I mean we have inside and outside counsel, and there were a lot of people, especially Bob Kohn, focusing on that, and I trust Bob a lot. And I tell Bob, you know, "You tell me [147] what we should do," I delegate to him, and he tells me, but that's all.

Q. Well, was Bob Kohn equipped to decide whether or not there would be a major impact on your ability to continue to market Quattro if you took out the .MU file, the 1-2-3 compatible menus?

A. Bob has been in the industry for a long time and with the company for a long time and I trust his judgment.

Q. So you would defer to his judgment on the question of whether or not you would be able to market your program without 1-2-3 compatible menus?

A. I would.

MR. GREGORIAN: Talking about the marketing point as opposed to any legal point?

MR. GUTMAN: I have been for several questions now.

MR. GREGORIAN: Okay, fair enough, you can answer.

THE WITNESS: As a matter of fact, I probably would because Bob used to run international sales at Borland, as well as being the corporate counsel, so he's very much aware of marketing issues. Bob wore two hats at Borland.

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Civil Action No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.
Defendant.

APPEARANCES:

O'SULLIVAN, GRAEV & KARABELL 30 Rockefeller Plaza, New York, New York 10112.
For the Plaintiff.
BY: KERRY L. KONRAD.

FENWICK & WEST, Two Palo Alto Square, Suite 80, Palo Alto, California 94306.
For the Defendant.
BY: EDMOND C. GREGORIAN.

DEPOSITION OF DAVID E. KIERAS
TAKEN OCTOBER 7, 1991

* * *

[24] Q. Is there a generally accepted definition of the term?
A. I don't know that I've ever seen it defined anywhere.
Q. Do you consider yourself to be a computer scientist?
A. Yes, I would say yes.
Q. Do you have a degree in computer science?
A. No.
Q. Your degrees are in psychology?
A. That's right.
Q. What experience do you have with respect to the development of microcomputer software?
A. What do you mean by microcomputer?
Q. What is commonly called a personal computer.
A. Personal computer such as a PC or MacIntosh?
Q. Yes.
A. I've developed very little software for those particular platforms.
Q. Developed what?
A. Very little software for those particular platforms.
Q. When you say very little, how much?
A. A few small programs. I've supervised other people writing a few programs.
Q. What few small programs are you thinking of?
A. Like a program to translate between file formats, [25] supervised people developing software in SuperCard.
Q. What do you mean by that?
A. Excuse me?
Q. What do you mean by that?
A. What do I mean by —
Q. Developing software under SuperCard?
A. For my research.

MR. GREGORIAN: That's not what he said. Object to the form.

Q. (By Mr. Konrad): What is SuperCard?
A. It's a HyperCard like program.
Q. And what have you supervised people developing using SuperCard?

A. Software for use in my research.
Q. Has any software program that you've worked on ever been commercially sold?
A. Commercially sold?
Q. Yes.
A. No.
Q. Have you ever worked for a personal computer software company?
A. I will have to ask you to say what you mean by worked for.
Q. Been an employee of or even consulted for.
A. Yes, I have a consulting arrangement with one, but [26] the arrangement was to help them develop their proposal which if funded I would be a formal consultant for. So I did spend time helping them develop the proposal.
Q. Proposal for what?
A. To develop some software.
Q. And what company was that?
A. Personal Bibliographic Services.
Q. What kind of software was this?
A. This was software to augment some data base retrieval operations.
Q. But you didn't work on an actual product?
A. That's correct.
Q. In any of your studies, have you attempted to analyze how individuals use personal computer software in the work place?
A. Excuse me?
Q. In any of your studies, have you attempted to analyze how people actually use personal computer software in the work place as opposed to under your laboratory conditions?
A. No, I haven't done any what are called field studies.
Q. Do you consider yourself an expert in the business of computer software.

[27] MR. GREGORIAN: Object to the form. What do you mean the business of computer software?
MR. KONRAD: Selling it, marketing it, running a company that makes software.

THE WITNESS: No, I don't.

Q. (By Mr. Konrad): Do you consider yourself an expert in what makes people buy software?

A. What makes people buy software?

MR. GREGORIAN: Object to the form, it's vague and ambiguous. Is that different from marketing or—I mean, if you're after something else, you ought to specify what it is.

Q. (By Mr. Konrad): Consumer behavior with respect to the purchasing of commercial software.

MR. GREGORIAN: Object to the form. You can answer if you understand the question.

THE WITNESS: Well, I would not call myself an expert on why people make purchasing decisions.

Q. (By Mr. Konrad): In any of your studies, have you attempted to analyze the extent or manner in which personal computer software users use macros?

A. What do you mean by my studies? Researchers do a lot [28] of different types of things.

Q. In any of your studies, research that you've done of any nature, have you analyzed the extent to which people use macros?

A. The extent to which?

Q. Yes.

A. It sounds like what you want to know is have I collected any data or done any analysis that would allow me to say how often people use macros.

Q. Correct?

A. No, I have not.

Q. Have any of your studies attempted to analyze the manner in which people use macros?

A. The manner, the reasons why they would use macros?

Q. Yes, the reasons why, how they do it in practice.

A. In practice?

Q. In the field.

MR. GREGORIAN: He said he didn't do any field studies.

MR. KONRAD: Fair enough.

MR. GREGORIAN: You have three questions in there.

MR. KONRAD: Fair enough.

MR. GREGORIAN: Two of which are different.

[29] MR. KONRAD: Fair enough.

Q. (By Mr. Konrad): In any of your studies have you attempted to analyze the manner, the way in which people use macros?

A. In any of my studies, pieces of research, collection of data?

Q. Correct.

A. Carrying on analyses.

Q. Correct.

A. I would say to some extent, to a small extent.

Q. Which studies are you thinking of?

A. Well, this is why I wanted to clarify what you meant by study.

Q. Well, what did you have in mind when you answered my question?

A. I've presented user interface concepts to classes and short courses before. The subjects of macros does come up as one of the important parts of usability and what the very broad principles are for insuring that macro capabilities are easy to use, and that it's important that they be available, and that's the small extent to which my research has addressed that.

* * *

[32] Q. (By Mr. Konrad): Is it your view that most computer scientists recognize human computer interaction as a distinct field of inquiry?

A. I don't really know whether they do or not.

Q. Would you agree that the field of human computer interaction has had difficulty gaining credibility among computer scientists?

A. Having difficulty, I wouldn't put it that way. I would say that it hasn't happened yet.

Q. Do you recognize the article "An Overview of Human Computer Interaction" from June 1990?

A. That's right.

Q. As one that you wrote?

A. Yes.

Q. Page 68 you wrote,"User interface design and other aspects of HCI also have a credibility problem within the computer industries and academic computer science." Is that a true statement.

MR. GREGORIAN: Give him the context. You're holding the article in front of you. He doesn't have it in front of him.

MR. KONRAD: It's there if you like.

[33] MR. GREGORIAN: Can you point to him where on this page?

MR. KONRAD: (Indicating).

THE WITNESS: Okay.

Q. (By Mr. Konrad): Is that a true statement?

A. Within the context, yes.

Q. What's particular about the context here that makes that true or let me put it a different way, in what context would that not be true?

MR. GREGORIAN: Object to the form, unintelligible.

THE WITNESS: I don't understand.

Q. (By Mr. Konrad): You say in the context it's true. In what context would it not be true, the statement you wrote here?

MR. GREGORIAN: Object to the form, it's unintelligible.

THE WITNESS: I don't understand how I can answer the question that way.

Q. (By Mr. Konrad): Can you imagine a context in which that statement would not be true?

MR. GREGORIAN: Object to the form. Just ask him what he meant by the context. Do you want information or do you want to play games and waste time.

* * *

[40] Q. And would it be fair to say that some computer scientists have a view of HCI that does not place it within computer science?

MR. GREGORIAN: If you know.

THE WITNESS: I don't know that I've ever discussed that with anybody. It's possible but—

Q. (By Mr. Konrad): Well, in the same article, in "An Overview of Human Computer Interaction" you wrote what is the relation between HCI and computer science. "Computer science should include HCI but generally does not." Is that a true statement?

MR. GREGORIAN: Why don't you show him the context.

MR. KONRAD: It seems very straight forward.

THE WITNESS: The context here is concerning the disciplines, in particular the academic disciplines. It's not generally the case that computer science does require for example their students to take a course in HCI, and one of the signs that a field would be considered as fully part of another is if courses in that were required, that's not necessary, but if that's happened, that's usually because the faculty in that department have [41] decided it's an essential part.

Q. (By Mr. Konrad): Well, you also wrote that, and I think you just told me that this would be true, but let me ask to make sure. You wrote, "Thus courses in human computer interaction or user interface design are not common in computer science departments." Is that a true statement?

A. I believe that is, yes.

Q. You wrote, "Where they do exist, they tend to be viewed with some skepticism as being soft or not really engineering." Is that true?

A. I have had some experience with that myself. That's the basis of that statement, and it seems to be a fairly common attitude, but not necessarily a universal one.

Q. You wrote, "The field of HCI generally has a problem in that it has not established its credibility as a technical dis-

cipline in main stream computer science." Is that a true statement?

A. Generally I would. What I would say though again is that that is changing. The ACMSIG for example exists on this field. New curriculum proposals with computer science are being sponsored by ACM as a field of study.

* * *

[46] Q. Did you take computer science courses during that time?

A. No, I didn't.

Q. Did you take any computer science courses at the University of Michigan?

A. At the University of Michigan, I'm pretty sure the only thing I did in the way of course work as far as I can remember is registered for a course in theory which I then later dropped.

Q. I'm sorry, which you then later dropped?

A. Later dropped.

Q. You said before that the issues involved in human computer interaction have been known since or have been dealt with in some way since computers began. Do you accept that persons active in the field of computer science before human computer interaction acquired a name as a field may have had understandings and experiences pertinent to that study?

A. Pertinent to that study?

* * *

[55] (By Mr. Konrad): Did you and Professor Olson collaborate on your declarations?

MR. GREGORIAN: Object to the form. What do you mean collaborate?

MR. KONRAD: Work together in any way.

THE WITNESS: With your meaning of collaborate, the answer is I did not. I did comment back to Mr. Hayes I think it was on drafts of her declaration.

Q. (By Mr. Konrad): Did you edit her declaration in any way?

A. What do you mean by edit?

Q. Make words changes, mark it up?

A. No. I marked places that I thought I should comment on.

Q. Have you seen the final version of her declaration?

A. No, I haven't.

Q. Are you aware that 29 paragraphs of her declaration are almost verbatim the same as yours?

A. I noticed that in the earlier versions, yes.

Q. What explains that?

A. They're both drafted by the lawyers.

Q. Are you aware that 31 paragraphs of your declaration are almost verbatim with that of David Little, **[56]** L-i-t-t-l-e?

A. I haven't seen his declaration.

Q. Did the lawyers do the complete first draft for your declaration?

A. Did they do a complete first draft?

Q. Yes. Let me try it a different way. Was there any portion of your declaration for which you did the first draft?

A. That I did the first draft, I don't recollect whether any of the wording in any of my earlier notes showed up in that first draft or not.

Q. Now, in the very first paragraph of your declaration you state, "I am the person in the Department of Electrical Engineering and Computer Science at the University of Michigan who teaches computer user interface design." Do you mean to suggest that you are the only person in the department who ever teaches issues concerning user interface design?

MR. GREGORIAN: Object to the form. That's not what it says.

MR. KONRAD: It says, "I am the person," and I'm trying to understand how much emphasis we should put on the word "the."

MR. GREGORIAN: Why don't instead of paraphrasing what it says ask is there anyone who **[57]** teaches user interface

design issues in any course which basically is the thrust of the question.

MR. KONRAD: I asked the question.

Q. (By Mr. Konrad): Do you mean to suggest by that sentence, Mr. Kieras, that you are the only person in the department who ever teaches issues relating to user interface design?

A. That's not the intent of that statement.

Q. To your knowledge are you the only person who ever teaches issues concerning user interface design in the department?

A. No, there is at least one other person that teaches things related to that.

Q. Is it your understanding that in courses on software development and practices, computer science generally, issues of user interface design are not addressed?

A. Perhaps some aspects of it would be, but I teach the only course that is currently offered or has been offered to my knowledge that is specifically about user interface design.

Q. Do you know whether the issues of user interface design are addressed in other courses within the department?

[58] MR. GREGORIAN: Didn't he say he knows of one course or the one course.

Q. (By Mr. Konrad): Leave aside the one other course. In any other course do you know?

A. It's possible, but I don't have any direct knowledge of it.

Q. Do you have a comprehensive knowledge in your view of everything that is taught in any every course in the department?

A. No, I don't.

Q. Are you in charge of user interface issues for the department?

MR. GREGORIAN: Object to the form.

THE WITNESS: I don't know what that means.

Q. (By Mr. Konrad): Has the faculty delegated you the responsibility for teaching issues concerning user interface?

A. There is no official action of that sort.

Q. In Paragraph 3 of your Affidavit you say you are collaborating with another professor of computer science on the design of user interfaces for advanced computer aided design systems?

A. Yes.

SPECIALLY CONFIDENTIAL,
SUBJECT TO PROTECTIVE ORDER

Volume I
Pages 1 to 162
Exhibits 1 to 14

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,
Defendant.

CONFIDENTIAL DEPOSITION OF W. FRANK KING, III a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Jessica L. Gonzales, Professional Shorthand Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Thursday, September 12, 1991, commencing at 10:00 a.m.

PRESENT:

O'SULLIVAN, GRAEV & KARABELL
(by Henry B. Gutman, Esq.)
30 Rockefeller Plaza,
New York, NY 10112,
for the Plaintiff.

FENWICK & WEST
(by Mitchell Zimmerman, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[79]

* * *

Q. Okay. In your judgment, familiarity with the 1-2-3 menus is a factor that influences purchases of 1-2-3; is that correct?

A. Yes.

Q. And do you agree the ability to run spreadsheets that contain 1-2-3 macros is a factor that influences purchases of 1-2-3?

A. Yes.

Q. And ability to run 1-2-3 macros is a factor that influences purchases of competitive spreadsheets also; is that correct?

A. Yes.

Q. There are upwards of 7 million people who have used Lotus 1-2-3; is that correct?

A. I don't think your number's right.

Q. Well, what do you think is the right number?

A. Maybe twice that.

Q. And so perhaps 14 million people have used Lotus 1-2-3?

A. That's twice that.

MR. GUTMAN: We could stipulate to the [80] arithmetic.

Q. And those people are familiar with Lotus' menus and the menu structures and macros by and large; is that correct?

MR. GUTMAN: Object to the form and lack of foundation.
MR. GUTMAN: You can answer.

A. They're certainly familiar with the menu structure. Not everyone uses macros.

Q. In your judgment is it true that most of the 14 million or so people who are familiar with the 1-2-3 menus would be resistant to changing to a different menu structure?

MR. GUTMAN: Object to the form. Lack of foundation, but you can answer.

* * *

[81]

* * *

Q. Let me start from another point. Are you **[82]** aware of any evidence that people today—let me start again. Are you aware of any evidence that today people buy Lotus 1-2-3 in part because of its menus?

A. Oh, for sure.

Q. What is that evidence?

MR. GUTMAN: Object to the form. You can answer.

A. I talked to lots of customers.

Q. And what do they tell you about compatibility with menus?

MR. GUTMAN: Object to the form.

A. They say that compatibility with our menu structure is very important to them.

Q. Do they tell you anything else about the menus other than that the compatibility with menu structure is very important?

MR. GUTMAN: Object to the form.

A. The question is so broad the answer is Yes, they tell me lots of things.

Q. Do you hear from customers that any aspect of the menu structure other than compatibility plays a role in their decisions to buy 1-2-3?

MR. GUTMAN: I'm sorry, what do you mean by **[83]** "compatibility" in that context? Do you mean macro compatibility, or do you mean menu compatibility, keystroke compatibility?

Q. What do you understand "menu compatibility" to refer to?

A. The words, what they mean, how they're spelled, how they're laid out, the sequence, the hierarchy.

Q. And what do you understand "macro compatibility" to refer to?

A. There is a macro definition, language that the product supports, and you can write a program in that language, you can execute it in 1-2-3, that defines its correct operation. If you can take that same program and execute somewhere else and you get the same result, then I would say those two systems are macro compatible.

Q. If you take the same file and run it in two programs?

A. I didn't say "file."

Q. Is that the same program?

A. I said the same string of characters.

* * *

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF ROBERT H. KOHN
Wednesday, September 11, 1991
VOLUME II
CONFIDENTIAL
ATTORNEYS EYES ONLY
TRIAL COUNSEL ONLY

Reported by: DARLEEN J. HENRY, CSR #3599

---

* * *

[227] A. I said that we would not have proceeded to spend millions of dollars in R&D and marketing to create expectations in our customers' mind expecting to have that feature or particular capability in the product.

Q. Well, is it your view that Borland has, in fact, spent millions of dollars to create an expectation in users' minds that

Quattro and Quattro Pro provide macro compatibility with 1-2-3?

A. I think not only Borland has spent millions of dollars, but customers have spent millions of dollars in their own time and investment in the Quattro and Quattro Pro line of products subsequent to our introduction to Quattro, and they have expectations about these capabilities that we said that we were going to provide.

Q. And in your view, Borland's customers have invested millions of dollars specifically in ways that depend upon the 1-2-3 compatibility in Quattro and Quattro Pro?

A. Yes.

Q. I take it you consider that macro compatibility to be a very significant feature of the product then?

A. Now that it's been shipping for nearly three and a half years.

Q. Did you consider it to be a very significant feature at the time Quattro came out?

[228] A. I don't know. I don't have any basis of judging whether something is significant or not at that time. That was something Philippe Kahn would have considered.

Q. Well, are you saying that Borland has spent millions of dollars in developing Quattro and Quattro Pro around the 1-2-3 compatible menu files?

MR. GREGORIAN: Is there a limit to these? He said what he said.

MR. KONRAD: He gave an opinion and I'm entitled to explore that opinion.

MR. GREGORIAN: I don't think it was an opinion. I think it was a fact.

THE WITNESS: I've already answered that question.

MR. KONRAD: Q. I'm not sure that you have. I'm sorry. I'm going to ask for an answer to it.

A. I've already answered the question.

MR. KONRAD: Could we hear it back, please?

(Record read.)

MR. KONRAD: Well, let me try again.

Q. You told me before about the millions of dollars spent investing and the like in creating expectations in [229] customers' minds; I'm trying to understand the nature of these expenditures.

Are you saying that Borland has spent millions of dollars in the development of the 1-2-3 compatible menu tree files?

A. No.

Q. Are you saying that Borland has spent millions of dollars in developing other aspects of Quattro and Quattro Pro specifically dependent upon the inclusion of the 1-2-3 menu tree files?

A. Given that "aspects" is a very broad term, the answer is yes.

Q. What aspects?

A. The aspects that I mentioned earlier, creating customer expectations about the product.

Q. Well, are you talking about the product itself?

A. And training people who are accustomed to use these features that they're expecting to get in marketing to these people who are expecting to continue to get this feature.

In making future versions of the products that were consistent with past versions of the products so that customers can continue to rely upon the expectations that Borland gave them from day one that we shipped the product.

[230] Q. Well, I really meant to focus on the product itself and not so much on the marketing and other things that you mentioned.

Are you saying that Borland has spent millions of dollars in developing parts of Quattro and Quattro Pro other than the 1-2-3 menu tree files in a way that depends upon the inclusion of those files in the products, specifically with respect to the product work?

A. You cannot just talk specifically with respect to the product because the product itself reflects customers' expectations. You develop products in order to meet customer needs.

Q. Are you saying that—strike that.

Are you aware of any particular expenditures in development of parts of Quattro and Quattro Pro other than the 1-2-3 menu tree files that would not have occurred if those files were not in there?

A. I'm not technically—

MR. GREGORIAN: Can I hear the question, please.

(Record read.)

* * *

[236] He referred to marketing expenses at least four or five times in this sequence of questions, and frankly, I don't remember the context, and I doubt that he does, so when you say when you referred to marketing expenses, you know, at what point?

MR. KONRAD: Q. Well, do you—

MR. GREGORIAN: Maybe one reference was broad and one reference was narrow, I don't know.

MR. KONRAD: Q. Is it your view, Mr. Kohn, that all the marketing expenses for Quattro and Quattro Pro relate to creating an expectation in the consumers' minds that the product provides macro compatibility with 1-2-3?

A. I know that if you remove some feature that a customer expects to have, you have wasted millions of dollars over a period of three years in marketing the product.

Q. Well, is it your view that if you removed the 1-2-3 compatible menu trees today, all the money spent in marketing Quattro and Quattro Pro would be wasted?

A. I'm not saying that.

Q. Well, how much would be wasted?

A. This is like hypothetical and—

MR. GREGORIAN: Objection.

THE WITNESS: —and I don't understand [237] how I can hypothetically talk about this. I'm trying to simply say that the company spent millions of dollars in development acqui-

sition, marketing expenses over a period of time to create customers' expectations about that product.

MR. KONRAD: Q. In specific, an expectation would be macro compatible with 1-2-3?

MR. GREGORIAN: Objection to the form. That's not what he said in specific. He said there was millions of dollars spent on marketing; that's what he said.

MR. KONRAD: Q. Yes, but the customers' expectations that you're talking about, in this context, you're referring to an expectation that it would provide macro compatibility with 1-2-3?

A. Look, we're talking about a hypothetical situation as to whether Lotus would have any objection whatsoever which they never gave us. I don't know what that objection might be.

I don't know whether it would be macro compatibility. I don't know whether it would be the existence of the 123.RSC file or a number of things or whatever; they may have asked us to change a word.

It's all speculative as to what they would have asked us to do.

[238] Q. I'm trying to focus on these marketing expenses.

Do you believe that Borland would have spent less money marketing Quattro and Quattro Pro if the product hadn't contained a 1-2-3 menu tree file or hadn't been macro compatible with 1-2-3?

MR. GREGORIAN: Object to the form.

It's hypothetical.

It's compound.

I don't think it gives sufficient facts to answer.

THE WITNESS: I don't understand—

MR. GREGORIAN: If you can take a stab at it—

THE WITNESS: I don't understand the question because it's compound. I don't know what questions I'm supposed to be answering.

MR. KONRAD: Q. Well, is it your view that Borland would not have spent as much money marketing Quattro and Quattro Pro if the product hadn't contained 1-2-3 compatible menu tree files?

MR. GREGORIAN: Object to the form.

You know, come on. I mean, you know, they might not have brought out the product at all if this objection had been made timely.

[239] There's a whole range of possibilities here, and, you know, I really strongly suggest to you that you can—I know that you're cleaver enough and capable enough to spend all day on this line of questioning; I don't think it's productive.

MR. KONRAD: Mr. Gregorian, if you'll stipulate right now that this witness will not submit any affidavit in which he offers any view or opinion as to what Borland would or would not have done had Lotus taken some action at any particular time before this lawsuit commenced, and if you will stipulate that he will not offer in that affidavit or any such affidavit any views or opinions or testimony concerning any amounts of money or activities or acquisitions or whatever, that Borland did supposedly in reliance upon what Lotus did or did not do or say, I will happily cut this short.

If you're not prepared to so stipulate, if you have any intention of offering that kind of testimony on your own behalf, you're completely off base in trying to deflect me from the examination right now.

All you're accomplishing is making it extremely unlikely in my view that we will conclude.

MR. GREGORIAN: No, I'm not going to enter into that kind of stipulation—

[240] MR. KONRAD: Then I see no alternative but to proceed.

MR. GREGORIAN: —on the subject that got me nowhere. The last time I did that, the witness I told you wouldn't offer declarations, you wanted to depose anyway.

MR. KONRAD: I'm not sure that's right. I'll be happy to discuss that off the record because I suggest—

MR. GREGORIAN: I'm not inclined to enter into that kind of stipulation.

MR. KONRAD: Why don't we—I have no alternative but to proceed. Why don't we take the break that you suggested.

MR. GREGORIAN: All right.

(Recess.)

MR. KONRAD: Q. Mr. Kohn, you do believe, do you not, that if Lotus had objected immediately when Quattro came out to the inclusion of the 123.RSC file, Borland could have found a way to revise the product and continue to market Quattro?

A. Yes.

Q. Do you have any reason to believe that if that had occurred, Borland would not have continued to spend millions of dollars in further development and marketing **[241]** of the product?

A. I have reason to believe that we would have continued spending money to develop and market the product.

Q. Do you have any estimate in your own mind as to how much more Borland spent in further development and marketing of Quattro because the 123.RSC file was in there than it would have spent if it hadn't been?

A. Again, I think this is hypothetical to ask, I just don't know how I could probably have a basis for determining the answer to that question given the fact that we did include the 123.RSC file.

Q. Returning briefly, I hope, to the specific point of marketing, do you have any estimate in your mind as to how much of the marketing expense incurred in relation to Quattro and Quattro Pro referred specifically to the inclusion of 1-2-3 compatible menu trees?

A. I have no basis for the determining how to apportion marketing expenses.

Q. I appreciate that answer and I may be able to wrap this up quickly. I'm going to try at a more general level and I'm telling you up front that I am to see if we can get out of this.

Do you have any basis for apportioning in your **[242]** mind how much more Borland spent in marketing Quattro and Quattro Pro because it had the 123.RSC files or the 123.MU files?

A. I have no basis for comparing something against something that didn't happen.

Q. You mentioned before the acquisition of Surpass—again, whatever you said before about it will speak for itself, I'm just trying to introduce that subject.

Is it your view that Borland would not have acquired the assets of Surpass if the Quattro product at this time had not contained an 123.RSC file?

MR. GREGORIAN: You mean assuming some Quattro product was released but without a 123.RSC file?

MR. KONRAD: Exactly.

MR. GREGORIAN: As opposed to the other alternatives. Okay. Fine. You can answer on that assumption.

THE WITNESS: Again, I don't have any basis for determining what we would have done if something—what we would actually have done had something turned out to be what was not the case.

* * *

CONFIDENTIAL, SUBJECT TO
PROTECTIVE ORDER
Volume II

Pages 2-1 to 2-95
Exhibits 10-19

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,

Defendant.

CONTINUED CONFIDENTIAL DEPOSITION OF THOMAS M. LEMBERG, a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Anne H. Bohan, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Wednesday, July 3, 1991, commencing at 10:00 a.m.

PRESENT:

O'SULLIVAN, GRAEV & KARABELL
(by Henry B. Gutman, Esq.)
30 Rockefeller Plaza,
New York, NY 10112,
for the Plaintiff.

FENWICK & WEST
(by Edmond C. Gregorian, Esq.
and Lisa A. Olson, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[2-58] (Document marked Lemberg Exhibit 13 for identification)

Q. Mr. Lemberg, I show you Lemberg Exhibit 13 for identification, a reprint of an article that appeared in *InfoWorld* September 3, 1990. Have you seen this article before today?

A. I don't remember seeing it, but probably I did, since I read that publication, and I certainly read articles about us and about this matter.

Q. Calling your attention to the paragraph that begins at the top of the second page of the exhibit, it says, "Excel has a macro translator that gives 1-2-3 macros comparable functionaltiy in Excel, Lemberg said. 'Lotus used that as an example of something that didn't violate our copyright,' he said."

Do you have any reason to believe you're quoted inaccurately in that respect?

[2-59] MR. GUTMAN: Object to the form.

A. I'm not certain that the first sentence is completely accurate. It's got the germ of accuracy, but I'm not sure—I don't believe I would have used the phrase "comparable functionality" in that context. I believe rather what I would have said is that Excel has a macro translator that runs many but not all 1-2-3 macros, but runs them in Excel. The phrase

"comparable functionality" appears to be the reporter's translation of what I just said into those words, and I'm not totally sure I think that's really a fair translation.

Q. Fair enough. Any other inaccuracies?

A. In the second sentence what I would have said, I believe—I don't recall the specific conversation, but what I would have said was that in the Paperback trial, we had discussions of macros, and Paperback's defenses about macro compatibility were used as an example, if memory serves, the Judge used them in his decision as an example of something that did not violate our copyrights, the translator facility that was in Excel that was discussed at trial.

* * *

[Lemberg Ex 13 7-3-91]

InfoWorld September 3, 1990

Copyright (c) 1990 IDG Communications, Inc.;

SECTION: INDUSTRY; Reaction; Lotus; Pg. 46

LENGTH: 524 words

HEADLINE: Lotus Litigation Sparks Corporate Resentment; Suit Viewed as a Way to Gain Market Share

BYLINE: BY BARBARA DARROW

BODY: If Lotus Development Corp. thought it could help its competitive standing in corporate America by suing spreadsheet competitors, it may have another thing coming, said users and analysts.

The perception that Lotus is using copyright infringement litigation to retain market share has sparked some ill feeling among large corporate users. Moreover, corporate customers express worry that Lotus might attempt to extend its copyright to keystroke sequences, and thus to customer-written macros.

"The lawsuit is pretty despicable," said Rich Minutillo, a senior automation analyst with CIBA-Geigy Corp., in Medfield, Massachusetts. It would be "terrible for the industry" if Quattro Pro had to be changed so it didn't offer 1-2-3 keystroke compatibility, he added.

The manager of a large chemical company said such a move could prevent corporate users from continuing to use the complex Lotus macros they've developed over the years.

"If someone tries to copyright the macros [we use in calculating] our budgets, there'll be hell to pay," he said.

Others dismiss that notion, claiming that U.S. District Court Judge Robert E. Keeton's finding in the Paperback case precludes pushing copyright protection that far.

But Jack McGrath, who follows the industry for his Hingham, Massachusetts-based at Max newsletter, said that Lotus' lawyers didn't help matters by "crowing that 'Slash Range Format' is now copyrightable. But it wasn't and the judge said it wasn't," he added.

"If you follow what Lotus is saying [about copyrightability] to its logical conclusion, there may be an issue," said one analyst.

Lotus itself said this is not an issue.

"We believe the customer owns the macros that he or she creates," counsel for Lemberg, corporate counsel for Lotus. "Our quarrel is not with customers at all—it happens to be with a vendor or two we think have violated our copyright."

Excel has a macro translator that gives 1-2-3 macros comparable functionality in Excel, Lemberg said. "[Lotus] used that as an example of something that didn't violate our copyright," he said.

Other companies are unperturbed by the skirmish in the spreadsheet world. Baxter Healthcare of Deerfield, Illinois, uses 1-2-3, Releases 2.01, 2.2, and 3.0 and Excel. And though most users who wanted presentation-style graphics upgraded to Excel rather than to 1-2-3 3.0, none cited Lotus' litigiousness as a reason, said Glen Jurmann, manager for office technology.

Corporate critics contend that seeking to quell look-alike programs is bad business because it bucks the trend toward making applications more uniform looking and, hence, easier to learn and use.

"Sure, you can't take Lotus code and put your name on it, but certainly if there's a close look and feel to [another program], I have no problem," said Ivan Brass, vice president and director of information services for Manufacturers Hanover Trust, in New York.

Regardless of how the Borland case comes out, Lotus may be losing in the court of public opinion. "The Borland lawsuit cost them a lot of goodwill in and out of the industry," the analyst said.

GRAPHIC: Picture, no caption

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF DAVID E. LIDDLE, Ph.D.
Thursday, October 10, 1991
CONFIDENTIAL
ATTORNEYS EYES ONLY
TRIAL COUNSEL ONLY

Reported by: DARLEEN J. HENRY, CSR #3599

---

* * *

[7] Q. You are being paid to participate as an expert on behalf of Borland in this case; is that correct?

A. Yes.

* * *

MR. GUTMAN: Q. I'll show you what's been marked as Exhibit 1, Dr. Liddle. Is that a statement you rendered to Borland with respect to your services in this case?

A. No, it's a statement I rendered to Fenwick & West.

* * *

[46] Q. I have a question or two about some of the things in your background as reflected in your declaration.

It says here that you received a bachelor degree at the University of Michigan in electrical engineering.

A. Yes.

Q. Did you take any computer science courses then?

A. They weren't called computer science courses then, but I did take some.

Q. Do you know Bernie Galler?

A. I have met him, of course; he was a well-known professor at the time that I was there.

Q. Did you ever take any courses with him?

[47] A. I didn't directly have a course from him.

Q. Are you familiar with his reputation?

A. Sure.

Q. And could you describe it for me.

MR. GREGORIAN: His reputation generally or his reputation for some specific character trait?

MR. GUTMAN: His reputation for expertise in computer science.

He said he was a well-known professor at the time.

Q. Is it fair to say, sir, that Professor Galler is a well-recognized expert in computer science?

MR. GREGORIAN: You're asking now, though, you're not asking at the time, right?

MR. GUTMAN: I'm asking now.

If it's changed over time, I'd certainly be happy to hear about that.

MR. GREGORIAN: Your question is now?

MR. GUTMAN: Yes, now.

THE WITNESS: Ask me the question again.

MR. GUTMAN: Q. Is it your understanding, sir, that Professor Galler is a recognized expert in computer science?

A. It's my understanding that Professor Galler is a recognized expert in system design and system **[48]** programming of a certain class, and at the time that I was, for example, at the University of Michigan, he was well-known for working on a very complex and very modern host-based time sharing operating system, okay.

And that was work that I admired and was familiar with, and that Professor's Galler's reputation—and I don't mean to split hairs about what is and what is not computer science—has generally preceded the time in which computer science has developed a lot of its fine grain structure as a discipline.

So I'm really trying to give you the most honest answer and not in any way say anything bad about Professor Galler.

Q. I understand that.

A. To make the point, for instance, I don't know of Professor Galler having any particular expertise in computibility theory, although I'm sure he's familiar with it.

I do know him to have worked a lot in the area of system programming.

I do know him to have done some imaginative and rather modern kind of industrial application.

So across the broad field of computer science, I would say that there are a number of areas that I **[49]** don't know him to be a recognized expert, but it's certainly the case that's he's been a long recognized educator in computing.

* * *

[80] Q. Aren't there spreadsheet users, Dr. Liddle, who identify the horizontal arrangement across the top of a spreadsheet

screen of the words, "Worksheet," "Range," "Copy," "Move," et cetera, in that order to a particular spreadsheet product?

A. You're asking me for my guess about that or my **[81]** opinion about that?

Q. If you have no opinion or you have no basis for an opinion, you can say so, but the question is, isn't it the case that to some number of spreadsheet users seeing those words in that order horizontally across the top of the screen is something that they identify with a particular spreadsheet program, namely, Lotus 1-2-3?

MR. GREGORIAN: You're asking without regard to the manner in which they are presented, just the words and their order horizontally?

MR. GUTMAN: I'm talking about those words horizontally on a spreadsheet grid, on a spreadsheet screen.

THE WITNESS: Yes, I'm sure there are such people. They are called Lotus 1-2-3 users, and I'm sure they identify them that way.

If you went to Quattro Pro users who haven't been 1-2-3 users, it's almost certainly the case that they believe that those are—that those are commands from Quattro Pro somehow if they have seen them displayed in that way; but I would agree with you that it's likely the case that people who have used 1-2-3 associate that sequence of commands in that order as do Macintosh users associate—

Q. File, Edit, et cetera?

[82] A. Precisely. I think that's right.

Q. And that is part of the product's identity in that sense, right?

A. Yes, it certainly is.

MR. GUTMAN: It's 12:15, you wanted to break for lunch, Ed?

MR. GREGORIAN: Yes.

(Luncheon recess from 12:15 p.m.)

* * *

[107] Q. Did you make any effort to determine whether the same sequences of keystrokes that would accomplish a certain result in Lotus 1-2-3 would in that mode of Quattro Pro accomplish the same result?

A. Yes, I did.

Q. And what was the result of that?

A. In all the cases that I tested, it did accomplish the same result.

Q. Did that strike you as coincidence or the result of an intentional design decision by Borland?

A. Oh, it was clear—it was clear to me that it had carefully been made so that those sequences of commands would produce the same results.

In fact, I also made some simple macros command on both to make sure that they would work in the same way in both, and, of course, those two things are inevitably linked.

Q. Is it your testimony, sir, that absent the existence of the 1-2-3 compatible menus, Quattro Pro would not be able to run 1-2-3 macros?

A. It's not my testimony that it would be unable to run previously constructed 1-2-3 macros.

It is my testimony that you wouldn't be able to modify or change or upgrade or debug previously constructed 1-2-3 macros.

[108] Q. So it's possible to run previously constructed macros without duplicating the 1-2-3 menu structure in the new program, right?

MR. GREGORIAN: Now you're asking a general question as opposed to one reference to Quattro Pro?

MR. GUTMAN: Yes.

THE WITNESS: Yes.

Let me, again, give you an expanded answer and tell me if it fits your need.

There's no facility that I know of to do that in 1-2-3—I'm sorry—in Quattro Pro, but when you say it is possible to—let me try and make a statement, see if that satisfies your need rather than answer directly to your question.

You could write a program which either interpreted a macro or failed, just said, "Oops, problem. Please go away and leave me alone," okay, that could parse well-formed macros and produce a final result, okay.

That's sort of Turing's definition of a computer, okay, is that it can recognize a given sequence of symbols. So it's clearly the case that one could make such an interpreter or compiler, whatever you want to think of it, and there's no question that you could do that.

[109] MR. GUTMAN: Q. I know you've read Bernie Galler's declarations, but in his declarations Bernie describes the fact that you could, if you wanted to, create such an interpreter of macros or you could do what Microsoft did with Excel, and do essentially a one time translation, and both of those are technically doable, right?

A. They're both certainly technically doable.

The pity is that it isn't what the customers want. In other words, you can do that technically, but that's only a technical stunt because the customers don't say that I want to use it one time and translate everything from 1-2-3 macro language over to a new one which now I can only use and which doesn't translate back.

What they say is, I have an asset base of all this stuff that I've built; I want it to be able to be interpreted by all the different spreadsheets that I buy. I want those spreadsheets to be able to modify those macros in a common way. That means that all of those different spreadsheets, I'll still be able to use them.

Q. So that argues for interpretation rather than one time conversion?

A. Yes.

[110] Q. But interpretation is technically possible, right?

A. Not only is it technically possible, that's exactly what's done. That is what Quattro Pro does.

In fact, the interior structure of Quattro Pro is only an interpreter anyway even in its native mode as opposed to the 1-2-3 mode.

* * *

Q. And by your reference to interpretation in the internal works of Quattro Pro, are you referring to the fact that you have so-called menu equivalent commands that you can take commands from either the native Quattro Pro menus or from the 1-2-3 menus and they have a common identifier in terms of a menu equivalent command?

A. That's essentially what I'm referring to.

* * *

[116] Q. Which is as to Quattro Pro, Release 1.

A. And Quattro.

Q. So it wouldn't —

A. Correct.

MR. GREGORIAN: What time is your call?

THE WITNESS: 2:30.

MR. GREGORIAN: He has a call he needs to take at 2:30.

MR. GUTMAN: I was about to start something. Let me see if I can do it in a question or two.

MR. GREGORIAN: All right.

MR. GUTMAN: Q. Paragraph 63 of your declaration, which is on page 20, seems to suggest that the words in 1-2-3 menus get incorporated somehow in macros and 1-2-3. That isn't really correct, is it, the words?

MR. GREGORIAN: Take time to read the paragraph because there's an assertion in that question about what you're saying in there that may or may not be accurate.

THE WITNESS: If I read the textual form of the macro, it's, of course, just a series of characters, most of which are letters and some are tildes—

[117] MR. GUTMAN: Tildes, T-I-L-D-E-S.

THE WITNESS: And other things.

Those characters are not unique in the commands that they represent without knowing what each character means in terms of the word it represents.

For example, "C," if it appears at the beginning, means "Copy," but if it's preceded by "R and F," it means "Currency."

So the keystroke "C," unbeknown to the word "Currency," isn't sufficient, okay.

In other words, the information contained in the keystrokes is not the same as the information contained in the words. You need the minimum of the words to construct or to decode the macros.

MR. GUTMAN: Q. Well, you need more; you need both the word and the hierarchical placement.

A. Yes, it is the ordering because it's what's called "context sensitive parsing." So for the human user either to construct or debug or change those macros requires the words and the hierarchy in which they fall.

Q. Or a good memory of that. If you have the hierarchy of the letters, you can do it without the words; but the words provide information if you don't remember what the "C" means after "R and F," isn't that

* * *

[119] you don't have a grammar for interpreting those strings of characters.

* * *

Q. Am I also right in understanding—and I believe that you say something to this effect in your declaration—that for

the various words used in the 1-2-3 menus, there are other words that one could choose that would convey the same meaning?

A. Convey the same meaning to a user. I think what I said is you could design—I forget what I said.

Anyway, you could design something where you **[120]** gave meaning to the term "Duplicate" instead of the term "Copy," and chose that as the consistently used word to mean that function, right.

In other words, although there are narrow groupings of words, in most cases the choice of words in the first instance is not unique.

Q. And, for example, range could be "Block" or something else instead of "Range"?

A. Right.

One could have designed 1-2-3 to say "Block," and "Duplicate," and—

Q. "Spreadsheet" instead of "Worksheet"?

A. (Witness nods head.)

Q. Isn't it also correct that if you were designing 1-2-3 over from scratch, not only could you use different words to represent each of those commands, but you could come up with various alternative ways of organizing those commands in the menus and submenu's and sub-submenu's; isn't that right?

MR. GREGORIAN: Is it just possible to do that, is that your question?

MR. GUTMAN: If you were starting over—let me give you a complete hypothetical.

If you were starting from scratch today, there was no 1-2-3, you were going to provide the same **[121]** functionality 1-2-3 does, but you were sitting down to design the menus, and assume for the moment we're using words instead of icons, which I know is not the way you would approach the problem.

A. That's right. It saves me the trouble of going through all that.

Q. Assume for the moment that you're doing it with words, isn't it right that as you sat down to design that set of menus, you would not only have the option of using different words to identify each of these commands, but you would have various options available to you concerning which of these commands you would put in the first level menu, which would be submenus, what the submenus would spring from; in other words, the entire structure, you would have the ability to make design decisions as to how to organize all of that?

A. Yes, there would be available to you some range of variability around each of those choices.

I was little bit unhappy that Professor Galler said that they were limited only by the English language or something like that because that seemed to me to be an excessively broad characterization. There aren't all that many choices, and you make the first set of choices, that limits the next set and the next set quite a bit.

[122] But it's certainly true that you have choices as to what the words are, what is, in a sense, left to right or sequential ordering of them at a given level, and then having made those choices, you have a smaller set of choices for which things go at the next level.

Those are to some extent determined by the first level, but for each word that you choose there, leaving out the ones already, blah, blah, blah, you still have a range of choices around all of those things.

Q. Again, taking as an example, Microsoft Excel with its menu addresses, essentially the same set of functions if not a greater set of functions than Lotus 1-2-3, but you have a series of menus and submenus and sub-submenu's that are substantially different; isn't that right?

MR. GREGORIAN: Object to the form of the question.

You've got the same set if not a greater set, substantially different.

I mean, you need to define some of those terms, I think, before he can answer that question.

MR. GUTMAN: Let me rephrase the question.

Q. In Microsoft Excel you have a spreadsheet **[123]** program which includes at least the basic functionality of Lotus 1-2-3 Release 2.

MR GREGORIAN: Why don't you stop there and see if he agrees with that.

MR. GUTMAN: Q. Isn't that right?

MR. GREGORIAN: If you know.

THE WITNESS: Well, what release of Excel are we talking about?

MR. GUTMAN: Q. 1 point anything, anything before 3.

A. And you said that has at least the functionality of?

Q. 1-2-3.

A. 1-2-3 Release 2.0?

Q. Yes.

A. Yes.

Q. Isn't it also correct that the menus of that product, "File," "Edit," et cetera, with the various submenu's and sub-submenus running from it—

A. Yes.

Q. —are substantially different from those of Lotus 1-2-3?

A. Yes.

* * *

[124] Q. Isn't it also the case that not only could you find other words for the words in the 1-2-3 menus, but if you were so inclined, you could replace those words with icons?

A. In order to answer the question, I need to go back to what it is you're asking me to design.

Q. We're designing a spreadsheet.

A. For manual use?

Q. You mean for keyboard use as opposed to mouse use?

A. No, I mean for human use as opposed to programmatic or macro use.

Q. Yes, I'm talking about human use.

A. Understand that the point that I'm on is you are saying isn't it true that you could do that and isn't it true that you could do that, and I want to be sure that you understand that I'm answering that in the context of whether or not us producing a product with a cockpit from which I could fly, so to speak, which I, as an active user—

Q. Yes.

A. —could do these things?

Q. Yes.

A. We have specifically excluded any requirement so far associated with executing anything, quote, like **[125]** macros or anything programmatic.

Q. We're not talking about how you would go about addressing macros written using some other program.

A. Or any language.

Q. For the moment, we're not addressing whether this product even has a macro capability.

A. That's my point, so I don't want—the term spreadsheet to many people presumptively means it includes that kind of capability.

Q. For the moment let's assume that we're just talking about a spreadsheet that doesn't have macro capability.

A. In that case, I would agree—

Q. The words could each be icons?

MR. GREGORIAN: Let him finish his statement before you prompt him with your long prompts

THE WITNESS: That was, in fact, what I was about to say.

That being the case, I will agree that I could have used iconic or graphic or spatial controls rather than the specific selection of textual words within some range of the expressiveness of icons.

Now, let me point out that it wouldn't be a one-for-one mapping. The user interface would have to be profoundly different since the semantic distinctions **[126]** like "Range" and so on and are very, very hard to capture in an icon that's at all

meaningful or on a button that's at all meaningful by solely graphical means.

That's the reason why most buttons are ultimately labeled "Text" anyway since most of the abstraction associated with office applications don't lend themselves very well to iconography.

MR. GUTMAN: Q. So that's why both in your work at Metaphor and the Apple Macintosh, I understand that t. ere are design decisions involved in what word and whether it's in the icon or under the icon?

A. Yes.

Q. But most of the icons end up being labeled since the images are not self-explanatory?

A. Right.

Q. And the word conveys meaning beyond picture?

A. Right.

Q. In your opinion, sir, is the inclusion of the 1-2-3 compatible menus in Quattro Pro something that's important?

A. Yes.

* * *

[130] that successive versions of 1-2-3 also have that 1-2-3 1.0 does not have.

And the only way that it's at all civil in my view—civil is not a good word—at all comfortable or manageable to use the 1-2-3 compatibility mode in Quattro is pretty much if you just stick to the same little range of commands that you might be using in 1-2-3.

The addition of the Quattro Pro capabilities into the 1-2-3 compatibility mode, in my view, is very ugly. That is, it's hard to use.

Q. But doesn't having the 1-2-3 mode there give the user who already knows 1-2-3 the ability to get up and running on the new program immediately and then gradually broaden one's use to pick up the new functionality that was added in Quattro Pro?

MR. GREGORIAN: You mean can one do that, is it that possible or likely or what?

MR. GUTMAN: Q. You can answer the question. I think it's clear enough.

A. If I'm a familiar 1-2-3 user, if I read the manual as to how to bring up Quattro Pro in 1-2-3 compatible mode, I will be able right away with only a minimal amount of relearning to do certain things that I did in 1-2-3.

[131] It's not you can't instantly do it, but it's not—doesn't require very much learning to understand that.

For example, it takes you a little while to catch onto the fact that you're not going to see the going forward choices, but you are going to see all the nested things that you've just done, and you are going to see the state variables and so on.

So, fairly quickly, you can right away do whatever was the last thing you were doing in 1-2-3.

Q. And it's not very difficult, is it, to figure out how to get the program to run on 1-2-3 mode?

A. No. Well, it's fairly clear in the documentation.

Q. And you can set that as the default so that henceforth it will boot up that way, right?

A. Yes.

Q. As as general principle, do you think it's bad to cause users to have to learn new programs?

A. As a general principle what I do think is that there needs to be value to new functionality to justify learning a new program; in other words, I don't think it's good gratuitously to change programs horizontally for only the same degree of functionality.

* * *

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF JOANNE HIROMI LIN
Friday, September 6, 1991
CONFIDENTIAL
ATTORNEYS EYES ONLY

Reported by: DARLEEN J. HENRY, CSR #3599

---

* * *

[49] Q. Was the 1-2-3 program in QA at the time?

A. Yes.

Q. Do you have a copy?

A. Yes.

Q. Did you look at it in testing that file?

A. Yes.

Q. What do you recall in that respect?

A. We would write macros within 1-2-3 using the first letter key stroke, execute them against Quattro to make sure they ran.

At one point we also had two machines set up where we walked through the menu tree.

Q. When you say you had two machines set up, you mean you had two machines side by side, one with the developmental version of Quattro and one with 1-2-3?

A. Correct.

Q. When you say walk through the menu tree, what do you mean?

A. Pick the first item and then make sure that menu went to the same menu that 1-2-3 went to.

Q. You say "pick the first item," you'd hit, for example, the worksheet command in Quattro?

[50] A. Right, and you'd get a menu.

Q. Another menu which begins with global and it goes on with other commands?

A. Correct.

Q. And then you'd look at 1-2-3, pick worksheet, and then make sure that the same sub-menu appeared?

A. Correct.

Q. Do you recall that kind of testing finding anything that was out of order?

A. No.

Q. Did anyone help you in that particular testing, just the testing of the menu tree?

A. Yes.

Q. Who helped you?

A. The QA group.

Q. Anyone in particular within that group is what I meant.

A. No.

Q. Do you recall at any point, before Quattro came out obviously, anyone suggesting to you in any way that you shouldn't look at 1-2-3 software or 1-2-3 manuals in connection with work on the 123.RSC file?

A. I don't recall.

* * *

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

CIVIL ACTION NO.
87-0076-K

LOTUS DEVELOPMENT CORPORATION

Plaintiff and Counter-Defendant

vs.

PAPERBACK SOFTWARE INTERNATIONAL,

Defendant and Counter-Claimant;

and STEPHENSON SOFTWARE LIMITED,

Defendants.

And Related Cases.

DEPOSITION UNDER ORAL EXAMINATION
OF JIM MANZI VOLUME II

---

* * *

[323] First of all, what language is the source code for Lotus 1-2-3 written in?

MR. GUTMAN: Mr. Bezjian is out of the room.

MR. ERSPAMER: He's not here.

MR. GUTMAN: Is that specially confidential or is that something that's been disclosed?

A What version are you talking about?

Q Let's start with 1A.

A Most of it is in assembler. Most of 2 is in assembler.

Q Has the source code ever been translated in the C language?

MR. GUTMAN: What do you mean, by Lotus or somebody else?

MR. ERSPAMER: By Lotus.

A We're in the process of writing a new version of product that we have set publicly. I believe we've set publicly, and I don't care if it's particularly confidential. It's being written in C, which would be the fundamentals for release of 1-2-3 C.

* * *

[412] A. I think, as I said earlier when we were having discussion about macros, that it is a serious issue to our customers on a list of a dozen kinds of issues of this magnitude for us to be asking them to do anything that would render work that they had done previously on our products obsolete.

So we strive as a company to provide a smooth transition path from one product to another and actually, in many ways, from one product to another family of products.

Q Is it Lotus' position that other spreadsheet manufacturers cannot furnish programs that are capable of running macros, Lotus 1-2-3 macros developed by Lotus' cutomers?

MR. GUTMAN: Could I hear that question back?

(The following question is read back:

"Question: Is it Lotus' position that other spreadsheet manufacturers cannot furnish programs that are capable of running macros, Lotus 1-2-3 macros developed by Lotus' customers?")

MR. GUTMAN: I'll object to the form of the question.

[413] You can answer it if you know what he's talking about.

A We're prepared to compete with anybody. There's a new product that has just come to market which I believe translates or runs Lotus macros. It's called Microsoft Excel.

As far as I know, we have no issue of that product.

Q To follow up on your answer, if you remove the existence of a translator from the process, would you have any problem with that in a spreadsheet program that directly processed Lotus 1-2-3 macros without translation or the use of a translator?

MR. GUTMAN: I object to the form of the question. Are you assuming that you are or are not violating Lotus' copyright?

A I don't understand what you mean by saying, would you have any problem? Meaning psychic, emotional, financial? What are you talking about?

Q Well, do you believe it would violate Lotus' rights?

A Which rights?

Q The intellectual property rights?

A I believe that the macros that people develop using our products are their property.

* * *

SPECIALLY CONFIDENTIAL,

Volume I
Pages 1 to 236
Exhibits 1 to 11

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

vs.

BORLAND INTERNATIONAL, INC.,
Defendant.

CONFIDENTIAL DEPOSITION OF JIM P. MANZI, a witness called on behalf of the Defendant, taken pursuant to the Federal Rules of Civil Procedure, before Anne H. Bohan, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at the Offices of Hale and Dorr, 60 State Street, Boston, Massachusetts, on Thursday, August 22, 1991, commencing at 10:00 a.m.

PRESENT:

O'SULLIVAN, GRAEV & KARABELL
(by Henry B. Gutman, Esq. and
Kerry L. Konrad, Esq.)
30 Rockefeller Plaza,
New York, NY 10112,
for the Plaintiff.

FENWICK & WEST
(by Edward C. Gregorian, Esq.
and Stuart P. Meyer, Esq.)
2 Palo Alto Square, Suite 800, Palo Alto,
CA 94306, for the Defendant.

* * *

[33] A. On CompuServe, right, that was one.

In terms of their awareness, I would have to conclude that they were aware on the grounds of the firestorming in the press that followed our filing of the lawsuit against Paperback, and if you did a frequency analysis of who was quoted more often than anyone else, I think it would probably be Mr. Kahn. This was not a silent little filing as far as our industry goes, so we also communicated our position through that open letter, I would assume that awareness was furthered by that.

And this was a topic of discussion, I would assume, I don't know that directly, but I would assume this was a topic of discussion, since I had seen evidence that he, at a minimum, was aware.

Q. "He" being?
A. "He" being Mr. Kahn.
Q. Anything else that you believe put them on notice?
A. No, nothing specifically.

* * *

[127] So in its totality, it does include all that, but the piece that I would describe as sort of the diamond in the crown jewels has to do with menus: menu structure,

macros, key strokes, those kinds of things. People love the art that went into defining and making that.

Q. And there's what, seven million people out there who have used the product?

A. There's more than that, there are $7^1/_2$ million legal users, there's some ungodly number of pirates of one form or another.

Q. And those people have through usage become accustomed to the menus and the menu structure and the macro language; is that fair?

A. Extremely fair.

Q. And as a result of becoming accustomed to it, they are somewhat resistant to changing to a different menu structure or macro language; is that right?

MR. GUTMAN: Object to the form and lack of foundation, but go ahead.

A. They are not only accustomed to it, they have built their businesses or their work lives around our product in many instances, and at the **[128]** same time a lot of people just out-and-out love the product and the way the product articulates or in a sense speaks through its menu structure to the user.

* * *

[173] Q. Take a look at one more item, if you would, let me just get it for you. The item that I'm going to ask you about is preceded by a list of what I infer are Version 2.0 features, some of which are asterisked with a 1, and the Footnote 1 states, "These features may be implemented if menu changes do not largely affect compatibility."

A. Yes.

Q. Do you have any knowledge or understanding as to what that refers to?

A. No, I don't have a recollection. I don't understand what that is referring to either, where the 1 is referencing.

Q. Well, it's on a previous page.

A. It's on a previous page.

Q. There are a couple of them that are indicated on Page 25712, the third item, the fifth **[174]** item.

A. Yes.

Q. Jog any recollection on your part?

A. No. But again, if you said, What are the important issues that we worry about in product design on an ongoing basis?, I think I already testified that compatibility was at the top of the list. We actually had an experience around compatibility with Release 2.0 which was not totally nourishing for the company, and it was after the release of Release 2.0 and the subsequent release of 2.01 where the importance of compatibility was firmly stamped on everybody's forehead as the single, unifying concept that we had to manage from generation to generation of our products.

Q. And what was the problem of compatibility, sir, on Release 2.0?

A. My recollection of the problem, the 2.0 to 2.01 issue was that—and this is the level of attention that has to be paid and we do pay—is that we were about 99 44/100 percent compatible. We were incompatible on fundamentally undocumented features, and customers either found them or others that were documented, and we were off by 56/100 of a **[175]** percent in terms of compatibility, and it required us to re-engineer and do as fast as we could on a breakneck pace Release 2.01 to achieve 100 percent compatibility, because we made an awful lot of customers angry.

Q. In addition to undocumented features, was there a macro compatibility/incompatibility issue?

A. That I don't remember. But we had never had that problem since, because we pay such incredible attention to compatibility. I think we were talking about this earlier, it's the primary source of value for generation to generation.

Q. The point being that though it affected a small number of customers and was relatively inconsequential on a percentage basis, it turned out to be very consequential because it was a problem for specific customers, correct?

MR. GUTMAN: I object to the form.

A. I didn't say it affected a small amount of customers. The amount of wrath that was released as a function of that was something the company had never experienced to date.

Q. So the amount of wrath was out of proportion to the degree of incompatibility?

[176] A. I didn't say it was out of proportion to anything, it just stressed upon us from that point on the importance of 100 percent compatibility, that's the point I'm making. I wouldn't say it was out of proportion.

Q. Or as close to it as humanly possible; is that true?

A. 100 percent. I wouldn't say it's out of proportion. If customers scream all over the world about this item over and over again, it tells you it's consequential.

Q. It's important?

A. It's top-of-the-list important.

Q. One more item on this document, that footnote refers to menu changes affecting compatibility. Do you recall any discussion surrounding the development and release of Version 2.0 of menu changes other than the additional menu functions?

A. No, and I wouldn't have been involved in those discussions, that was Mitchell's purview and that's the way we ran the company at that point.

(Document marked Manzi Exhibit 7 for identification)

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Civil Action No: 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—against—

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF JUDITH S. OLSON, Ph. D.

Thursday, September 26, 1991

VOLUME I

Reported by: Kimberlee Cricks, CSR# 7473

---

* * *

[39] Q. You are being paid by somebody in connection **[40]** with your testimony here today, aren't you?

A. Yes.

Q. Who is paying you?

A. Comes through Fenwick & West.

Q. Have you had any contact with Borland directly yourself?

A. No.
Q. Talked to anyone from there?
A. No.
Q. Any written communication with them?
A. No.

* * *

[48] Q. What is your opinion?

A. I would not like to see the major aspects of user interfaces be copyrighted.

Q. Why not?

A. There is a lot of accumulated knowledge in the field that if they copyright some particular aspects of it, people have to design new things that are less good for the end user. I'm an advocate for the end user.

Q. When you say "major aspects of user interfaces," what are you talking about?

MR. GREGORIAN: Just give me a continuing objection to this line?

MR. KONRAD: Absolutely. No problem.

THE WITNESS: Major aspects of the user interface have to do with how to access the functionality, how to get the system to do what it was designed to do. That's what I mean. I would not like to see it restricted in any way. There are what I might phrase more artistic pieces to some interfaces I have seen that if they want to protect those, that's fine, like a logo.

* * *

[52] Q. Well, it is certainly possible, is it not, for a software developer to design a menu tree that uses any word to represent any particular function that the program provides to the user, isn't it?

A. Say the first part of your question again.

Q. Isn't it possible for a software developer to **[53]** use any word at all to represent any particular function that a spreadsheet program provides to users?

MR. GREGORIAN: Object to the form. Are you talking about theoretically possible? I mean there are a million examples one can think of where can it could be done, but it would make absolutely no sense.

MR. KONRAD: Parsing that might well be an important exercise.

Q. To begin with, I do mean just as a technical matter, you could pick any word to represent any command if you wanted to, if you weren't concerned about meaning; isn't that right?

A. That is true. Technically, you can map the function to any act that the system can read.

Q. And indeed there are sensible words that differ which are used in different spreadsheet programs to represent comparable functions, aren't there?

A. There are a few.

Q. In fact, in the Quattro product, for example, the word Block is used to represent something analogous in function to Range in 1-2-3; isn't that right?

A. That's correct.

Q. In your view, is there a best or right way to design a menu tree for a spreadsheet program?

A. There are a set of well-known principles that **[54]** will narrow down the set of possible words and organizations.

Q. Some of those principles are just the principles of the English language and meaning; isn't that right?

A. Some selection of the word. You want to pick a word that connotes the right meaning.

Q. To that extent, your constraint is the same constraint that anyone faces when they're trying to communicate with language; isn't that true?

A. In the sense of trying to convey ideas with words, yes.

Q. Is it your view that with respect to computer software, there is such a limited number of words that make sense that if people could not use exactly the same set of words to represent different functions of a spreadsheet program, they'd be reduced to using senseless words?

A. I think there are—there's a small set of words that convey certain kinds of functions. It's not a large set. When

you're designing a whole interface, it's not just the word but the word in contrast to the other words around them, what their first letters are and how big they are. When you use all of those principles together, there's not a large set.

* * *

[56] MR. KONRAD: I'm focusing on the word choice at this point.

THE WITNESS: You're asking me if there is a way to measure how large that set is. I have not personally measured it, and I am not sure—there are so many other constraints here in design of a good user interface that I think it's not an exercise that I would like to go through.

MR. KONRAD:

Q. Can you think of any example or instance where there's only one word that you believe would sensibly convey the meaning of a particular function to a user?

A. No. But I can imagine there being a very small set of words you could use if you're going to look for short, familiar, easy-to-understand words that don't conflict with the first letter of the rest of the things on a menu.

Q. Well, the point about whether they conflict with other things in the menu, I mean that raises an additional consideration, doesn't it, of how you actually organize the menu commands and what sequence you place them in, doesn't it?

A. It matters about what's on—what are the **[57]** competing things on the same level or whether you choose to put that one on a second level.

Q. Absolutely. And my point is that it is—strike that. I'm not here to make points. I was trying to ask a question.

Isn't it true that when you consider that question of conflicting items beginning with the same first letter on a particular menu level, the designer has the flexibility to place different commands on different menu levels to try to avoid that problem?

A. Yes.

Q. Is it technically necessary for the words—strike that.

Why do the words in a particular menu level have to begin with different first letters?

A. One of the ways some pieces of software have been designed to make it easy for people to use is that they don't have to type the entire word but just the first letter. Very common principle in design of user interfaces; therefore, you want to have unique first letters so people don't have to type in the first two letters to make it unique.

Q. But it's not technically necessary that you use the first letter in the word as the one which would indicate what keystroke to type, is it?

[58] MR. GREGORIAN: When you say "technically necessary," I take it you're talking about from a computer science point of view as opposed to from a cognitive science point of view. That is, you can get the computer to respond to the command on a first letter basis or on some other basis. Is that what you're asking about? Because you go back and forth between technically necessary and computer-human interaction principles, and I want it to be clear which one you're talking about on any particular question.

MR. KONRAD: I don't like to sort of engage in this kind of thing. I'm not going back and forth between anything, any principles here or there, because it's just not what I'm doing.

MR. GREGORIAN: You're not doing that? What do you mean by "technically necessary"?

MR. KONRAD: What I meant is in the sense that that's the only way that a software program can be designed to work.

THE WITNESS: That is not the only way a piece of software can be designed to work.

MR. KONRAD:

Q. In fact, Excel, another spreadsheet program, or Windows, which is an operating system or operating environment, uses letters other than the first one in a word as the indicator of **[59]** which key a user should strike to invoke that command; isn't that true?

A. It is true.

* * *

Q. Why is it any constraint on a developer at all, if it's true that you don't have to use the first letter in a word, that the words in a particular menu have to begin with different first letters?

A. It is easy for the user to think of the word that invokes the function and having a simple rule that says once you think of the word, type in the first letter. It's a general design prescription in our field.

Q. It may be easier for them to do, but it's not the only way to do it, is it?

[60] A. No. It's not the only way.

* * *

[63] Don't those native menu trees in Quattro and Quattro Pro demonstrate that it is possible to design a new or different menu tree from an existing product that would **[64]** actually be better from the end user's point of view?

* * *

THE WITNESS: There are aspects of the user interface of Quattro and Quattro Pro in addition to the menus and menu hierarchy that altogether make it a superior interface in my opinion. It is possible in that constellation there is a new menu tree and there are some new words.

MR. KONRAD:

Q. But you do believe that just with respect to the menu tree, the native ones in Quattro and Quattro Pro are easier to use and easier to learn than the ones in Lotus 1-2-3?

A. I do believe that.

Q. Doesn't that demonstrate that you could create a new or different menu tree from an existing product that would be better for users?

A. It is possible to build—this is a demonstration that one can be built. The question is: How many?

* * *

[115] Q. Grouping menus according to that principle, you said if it's designed well— I'm just trying to—

A. There is a design prescription that says you should group things according to similar functions, and the example I just gave you of having editing things underneath the word called Edit is a reasonable—it's a good way to do things. There are a number of principles used to put together good menu design.

* * *

MR. KONRAD:

Q. Right before the break, Professor Olson, you mentioned there were a number of principles for good menu design. I think we've talked about one, the grouping of items in particular menus.

What other principles do you have in mind?

A. One of them is to put frequently used commands **[116]** high on the menu.

Q. By "high," you mean what?

A. So that by choosing that, you don't have to make many choices before the function goes. Copy is an example.

Q. In other words, to put it on a menu towards the top of the tree rather than down in some nested subbranch?

A. As a general principle, the higher frequency ones should be higher in the menu.

Q. And others?

A. Just common words that people will recognize. And often you'd want to use short words because you're trying to save what we call screen real estate, screen space.

Q. Others that you have in mind?

A. There is a principle that you would like to match the order of what you have to choose with the goal you have in your head. For example, you often construct a sentence in your head of what you would like to do. It has verb, object, and modifier in it. That's the way you think, at least in English. And if you can—if you don't—if you can do it in that order, that's a natural order.

Q. Are you referring there, in particular, to a verb, object, modifier sequence or generally to a **[117]** consistent sequence of that nature? I'm not sure I understand.

A. There are two principles, and they're mixed. One is to fit the order you have in your head, and the other is to be as consistent as possible.

Q. And the order that, in your understanding, is in the English speaker's head is which again?

A. That's fitting the order which you have in your head. That's the principle. You also have another principle, which is try to be as consistent as possible.

Q. I may not be understanding. Is there a particular order in terms of verb, object, and so forth that you believe is consistently the more appropriate one?

A. There is— now these are competing principles. One says that you should use the order you have in your head, and commonly in English what you think of is verb, object, modifier. "Put that there" is an example. Okay. Now you also want to be consistent. You want— if you're going to choose an order, whichever order it is, try to keep doing that throughout. Don't mix it up. That makes people stumble.

Q. Are any of the other principles in good menu design competing in some sense?

A. There are always constraints that you have to deal with, and you can't fulfill all of these principles **[118]** at the same time, typically.

Q. In other words, the designer has to make some tradeoff, depending on what their objective is for the menu design?

A. Yes.

Q. This Lotus 1-2-3 uses the verb/object form of construction?

A. It is not consistent. There are some commands that do and some commands that do not.

Q. Are you familiar with the term that Ben Shneiderman uses for a kind of interface, direct manipulation interfaces?

A. I know the term.

Q. To the extent I understand that— and correct me if I'm wrong— this would refer to interfaces such as ones where you

could actually point to something on the screen and make it do something.

A. Yes.

Q. Do those interfaces operate by putting the object first and then indicating the action?

A. We have to be clear about direct manipulation interfaces because that is not a word that is— has a single definition. There are graphical interfaces for which— that are similar to this. There are icons or objects on the screen, and you move them through gestures

* * *

[120] Q. You believe internally to the user, there is the same principle in application?

A. I don't know what you mean.

Q. In either case, the user is forming some mental sentence with respect to what they want to do, whether or not it's words in the menu or the manipulation of objects on the screen?

A. I believe they produce a mental sentence in their heads, and then, depending on what kind of system, they will enact.

Q. Are there any other other principles of good menu design that you have in mind?

A. There aren't any that I can think of right now. That's all.

Q. In your opinion, is there creativity in good menu design?

A. There is a lot of hard work, but it's more akin to engineering and building a bridge, given certain kinds of principles and scientific knowledge.

Q. Do you think there's any subjective quality to a menu design?

A. Well-known principles and you grind through them and have to make some judgments about tradeoffs. It's a craft.

Q. In making those tradeoffs, you don't always **[121]** have data available, and models and so forth can't always predict to you exactly what the appropriate choice would be; isn't that right?

A. The models and the principles cover a large domain. It's not always the case that you've got everything you need to make a choice, so you make a choice of which way you're going to do it.

Q. Making those choices, doesn't the skill and judgment of the designer come into play?

A. The skill and judgment, yes.

Q. Doesn't the intuition of the designer come into play?

A. Well, there are—if you mean by "intuition," that people have stored experiences that they can bring to bear general rules of thumb about how to do this that is brought to bear—if you mean something else by intuition, I don't know.

Q. Well, when users perceive a menu design, isn't it possible that they will simply like one better than another?

A. They will like on the basis of preference because you can learn it easier. "I understand it better. I can use it more easily when I do my work."

Q. But the user won't necessarily go through the process of articulating those reasons to explain why they **[122]** prefer one over the other, will they?

A. Sometimes they do. Sometimes they just say as a blanket statement, "I like this one better."

Q. When it comes to choosing words for menus that will convey the intended meaning best to the user, what engineering principle would the menu designer apply?

A. In choosing the word?

Q. Um-hm.

A. You think of the function that you want to tell the user, that "This is—This is what you can do. This is what you will get if you select this item." Then you think of the first word you think of and look up in a thesaurus for all of the other words that seem similar and make some choice based on some of the other principles.

Q. Well, in your view, is the meaning conveyed by words something that is susceptible to an engineering analysis?

A. There are analyses that say how many people recognize this word, which is the most frequent in common English. The dictionary says what it means.

Q. Well, can word choices for menus just simply be ugly or inappropriate?

MR. GREGORIAN: Object to the form.

THE WITNESS: Ugly? They can be misleading.

* * *

[124] people don't like the word Abort is not because of its association with the word "abortion," but it seems like a very severe reaction, that something will be stopped and stopped very fast without regard to anything happening around it, which is frightening to some people. They think they've broken something.

Q. The word "kill," that might have the same effect?

* * *

THE WITNESS: Kill is a word that people don't use anymore because some people, again, thought it was a severe reaction, not because they thought of blood and guts.

MR. KONRAD:

Q. Well, you spent a lot of time analyzing user interfaces. You've done a lot of work in it. You've worked on them yourself. Do you not think that there is any creativity in the design of the user interface?

A. I think there's a lot of smart cognition going on when you do it, what I would call human factors engineering, and I do human factors engineering. Would you say there's cognitivity in building a bridge? There's a lot of work and knowledge and skill. I'm not sure it's a creative endeavor.

* * *

[125] Q. Do you have any understanding that to acknowledge there is creativity in user interface design would have some negative effect on the side you're working for in this case?

A. I think the word is a key word. That my use of the word in everyday language is very different from our use of the word in this case.

Q. Which word?

A. Creativity.

Q. How do you define creativity?

A. How do I define creativity? I don't think I have a good definition of it just off the top of my head.

Q. Well, would you say it's a creative act when somebody writes a particularly good instruction book on how to build a boat?

A. To be consistent with what I have—how I have been using it, that is a case of design and good **[126]** engineering principles for doing that. I teach documentation and how to do things, and it has a series of principles, and if you follow those principles, it takes no creative leap to do that. You can just grind it out, and it will be very good.

Q. Is there any creativity, in your view, in writing a software program? By that I mean to code it.

A. It depends on what the software is doing.

Q. Why would it depend on that?

A. Software for some functions that have been around for a while is not creative at all. It's hard to make sure you have all of the pieces that you need; however, there is other software coding that is very creative because you develop new algorithms. You take a leap from what you have seen before into designing something very new.

Q. Do you equate creativity with invention?

A. I don't know. Is there a technical definition of "invention"?

Q. No. Well, there might be for purposes of the patent laws, but that is not really at all what I meant. I just meant it in English usage. You just said before you create something, there would be some leap perhaps in certain coding. I'm just trying to understand your views on this point.

[127] You think that what makes that a creative thing is there was some inventive leap that the person had to make?

A. I would go along with that. There's an inventive leap. As I'm saying, if you have a technical meaning you're trying to extract here, that, I don't understand. Maybe this is normal English I'm using here.

Q. You've made that clear.

Do you think there's creativity in making a map?

A. There are standard maps, and there are inventive maps that I have seen, so it depends.

Q. Do you think there's creativity in making up icons for computer software?

MR. GREGORIAN: What do you mean "making"? Designing them?

MR. KONRAD: Designing them. I didn't want to use a word that might be loaded. I actually tried to avoid it.

Q. But in whatever it takes to make an icon that would appear as a possible selection to a user in a software program, would that be creative?

A. I can imagine, again, some of them being creative and again some of them not.

Q. What would it depend upon?

[128] MR. GREGORIAN: In her opinion?

MR. KONRAD: Right.

THE WITNESS: Well, inventive leap, a new concept where I would go "oh" as opposed to "I know what that is" and just go.

MR. KONRAD:

Q. Well, does the fact that you can apply rules to aid in better making something tend to negate, in your view, whether or not it's a creative act?

A. Making of rules? You mean that somebody is applying principles, and there are things that are known that I'm using in order to—

Q. Yes.

A. Well, to me, the more there is of that, the less creative it is. It doesn't mean it's easy because sometimes rules take you in different directions, and you have to put them together.

* * *

[136] MR. KONRAD: Q. Don't you recognize that a large number of the commercially successful software products in the last ten years or so were created—or user interfaces were created without their designers following the kinds of models

or even necessarily being aware of the kinds of empirical data that you and your colleagues have been creating?

* * *

[137] A. I recognize that designers do not often use models and empirical data that come out of research laboratories and still can produce successful software. They are, however, using the principles that fall from those. Though they don't recognize it comes from a scientific base, they do evolve into the general practice. Mitch Kapor talks about some of those.

Q. Yes. But you don't have any personal knowledge or belief that he was consciously following any of the models or principles that have been scientifically derived, do you?

MR. GREGORIAN: When you say "personal knowledge," do you mean to include or exclude paperwork in the case?

MR. KONRAD:

Q. Is there anything in the paperwork to suggest that he was doing that? That he was aware of any of these models and following them consciously?

[138] A. He articulates—Mitch Kapor, in his declaration, articulates general design principles about frequent things being at the top and using common words.

Q. Do you have any information to suggest that came from anything other than his own intuition?

A. He does not say where he learned that.

Q. Do you have any reason to believe he learned it from studying these kinds of books and articles?

A. I have no idea where he learned it.

Q. Aren't you, to some extent in deriving these models, simply observing the behavior of people like Mr. Kapor and trying to derive principles from what they do, in fact?

MR. GREGORIAN: Observing Mr. Kapor's behavior?

MR. KONRAD:

Q. People like Mr. Kapor who actually went out and created products, aren't you just trying to derive principles from observing what they do?

MR. GREGORIAN: Object to the form.

THE WITNESS: I don't observe what Mitch Kapors do. That's not accessible to me. I see only their end product. I evaluate their end products in the laboratory, various forms of them, and sometimes alter them myself to understand what the underlying design

* * *

[173] Q. Are you familiar with anything else in any of the documentation that would suggest that a reason for the 1-2-3 emulation modes is because they provide familiar commands for users?

A. I don't remember explicitly reading that phrase a number of places.

Q. That would be an advantage in offering it, wouldn't it?

MR. GREGORIAN: What does "that" refer to?

MR. KONRAD:

Q. That having the 1-2-3 emulation modes would allow people to use commands they're already familiar with?

A. Yes.

Q. Now in your view, is it technically necessary to display the 1-2-3 menus in order to permit the execution of macros originally written in 1-2-3 in Quattro or Quattro Pro?

A. Macros that are imported have to be altered to a new situation, and to understand and debug what they are, you have to read what's on the menus. The menus do funny things.

[174] Q. When you say "You have to read the menus," you mean you have to look at the menu commands?

A. You have to look at the menu commands, and, in particular, you have to know whether after you had a carriage return, whether the menus go away and you have to start them again with a slash or whether it stays in the menu item and you just progress into the next letter or symbol that you want to invoke.

Q. Are there any particular conditions—strike that.

Is what you're saying that any—strike that.

I don't understand the conditions in which you're postulating the person using the macro and needing to debug it.

What circumstance are you envisioning?

A. There's a macro that was built in Lotus that is imported to a new spreadsheet, a new situation. In that new situation, I add columns or rows or change the names of things or put other values in another cell. Macro no longer does what I expect it to do. I have to change it. Some macros were not designed to be transportable easily. They were not built very well, but they exist.

Q. Is the only way to allow people to preserve their macros, even if it's just to permit them to debug them, to display a 1-2-3 menu tree in the operation of the **[175]** program?

A. There are many things in the display of the menus for the purpose of debugging that are not in the manual anywhere. For example, when you invoke a particular menu item, does the menu go away or not? Once I hit a carriage return, does it go away? Some do. Some do not. Some put onto the screen the first value if you're trying to specify a range, and others do not. You have to know what it really looks like to the user who first built it in order to alter it appropriately.

MR. GREGORIAN: Can you kind of bring this to a conclusion. We passed our time. You said five minutes.

MR. KONRAD: Three more questions.

Q. Couldn't Quattro or Quattro Pro provide a translation or conversion utility for 1-2-3 macros?

A. Not if you're going to use the macro in a new situation and have to change it. There are things that appear on the screen, and there are some that involve menu picks in the middle of it that you have to have the menu items to choose from in order to do this right.

Q. But if it's translated or converted, you could then do further debugging in the new macro language, couldn't you?

A. I don't know.

[176] MR. GREGORIAN: With the exception she's noted, are you asking a general question?

MR. KONRAD: Generally.

THE WITNESS: I don't know.

MR. KONRAD:

Q. Have you ever done studies about debugging of macros and the problems people would encounter in doing that?

A. No.

Q. Have you ever studied translation or conversion of macros?

A. No.

Q. Have you ever looked at Xcel's translation ability?

A. No.

Q. Are you familiar with the command in Quattro Pro 2.0 and 3.0 called Tools Macro Key Reader?

A. I know very little about it. It will translate in the—if I understand it, it will translate some macros that were written in Lotus into the Quattro command menus.

Q. Which won't it?

A. I don't know. I know there are some.

Q. Any that you know it won't? Can you describe them.

A. I don't have an example of one that it will

* * *

IN THE DISTRICT COURT OF THE UNITED STATES
DISTRICT OF MASSACHUSETTS

Civil Action
No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF JUDITH S. OLSON, Ph.D.

VOLUME II

Taken on October 15, 1991

APPEARANCES:

O'SULLIVAN, GRAEV & KARABELL,
(by Kerry L. Konrad)
30 Rockefeller Plaza,
New York, NY 10112,
For the Plaintiff.

FENWICK & WEST,
(by Edmond C. Gregorian)
2 Palo Alto Square, Suite 800,
Palo Alto, CA 94306,
For the Defendant.

* * *

[60] Q. Well, when you say that the command hierarchy of 1-2-3 is a system for performing numerical calculations, you don't mean it's some kind of mathematical algorithm for solving problems, do you?

A. The command hierarchy is not an algorithm for solving problems. It is a set of steps that you go through so you can do your work that involves numerical calculations.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Case No. 90-11662-IC

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—Against—

BORLAND INTERNATIONAL, INC.,
Defendant.

DEPOSITION OF JACK OSWALD

THURSDAY, AUGUST 29, 1991

CONFIDENTIAL—SUBJECT TO PROTECTIVE ORDER

Reported By: Victoria Jensen, CSR# 6393

---

* * *

[4] BY MR. KONRAD:

Q. Would you please state your full name and address for the record?

A. My legal name on my birth certificate is John Oswald.

Q. Do you go by Jack?

A. I go by Jack.

Q. Pretty much all the time?

A. All the time.

Q. And your address is, sir?

A. Current home address is 4750 Capitola Road, Capitola 95010.

* * *

[5] Q. By whom are you employed?

A. Borland International.

Q. What is your current position there?

A. Product Marketing Manager.

Q. And is that with respect to spreadsheet products?

A. That's correct.

Q. All spreadsheet products for Borland?

A. Effectively, yes.

Q. What do you mean by "effectively?"

A. My charter is for Windows products only, but since I'm the only Product Manager, I work on all spreadsheet products.

* * *

[14] Q. How long were you the Product Manager, a Product Manager for Quattro Pro?

A. What do you mean by "a Product Manager"?

Q. Well, I saw Mr. Davoust two days ago and he was a Product Manager for some period of time.

A. Okay.

Q. So, I didn't want to say the Product Manager. But were you a Product Manager—well, just for how long were you a Product Manager?

A. Since the time the product shipped?

Q. Yes.

A. I was Product Manager all the way through until I became Product Marketing Manager.

Q. In April of '91?

A. Right.

* * *

[47] Q. I understand. In the presentations or conversations that you have had with the press or customers, has the subject of the 123-compatibility come up?

A. Yes.

Q. Has this been something raised by the press or customer?

A. In some cases, yes.

Q. Some cases you have raised the point?

A. Yes.

Q. What do you recall saying about that?

A. That there were four specific levels of **[48]** compatibility.

Q. Can you describe those?

A. Yes.

A. The main one, initial one being file compatibility, ability to load a file. The next would be macro compatibility, be able to run the macros without change; and the third would be capacity, be able—the ability to load a spreadsheet file into a similarly configured hardware machine.

Q. You mean a spreadsheet of a certain size?

A. Right. Well, not of a specific size but that a spreadsheet created on say a particular machine, in 1-2-3, there is released 2.01 or 1A could be loaded into version of Quattro Pro on that same exact machine.

Q. Okay. Exact machine. And number four?

A. And the last would be keystroke compatibility.

Q. And by that you mean specifically what?

A. The ability to load a 1-2-3 similar menu tree.

Q. And do what?

A. And be able to use similar keystrokes with that particular menu tree.

Q. Have you told people in these conversations where its come up that Quattro Pro provides all those four levels of compatibility?

A. Yes.

* * *

[54] THE WITNESS: It's hard for me to answer that question, since that was not the intention of why that point was brought up.

BY MR. KONRAD:

Q. Of why you brought up the point of keystroke compatibility?

A. Correct.

Q. What was your intention again in mentioning that separately from macro compatibility?

A. The intention was to have the product perceived prior to purchase as a product that would be easy for them to adopt, but that once purchasing the product, that they would use the new system menus in the product because they would be far easier to use and learn to accomplish tasks using spreadsheet.

Q. Would another way of putting that to be as a migration tool?

MS. OLSON: Objection. What do you mean by "migration tool"?

BY MR. KONRAD:

Q. Do you understand what I mean by that?

[55] A. You mean migration.

Q. From 1-2-3 to your product?

A. Not necessarily as a migration tool, but as a barrier to consideration of purchase, to reduce the potential barrier consideration.

Q. Aren't you saying then that you would mention that so that people would perceive that they could begin to work on your product using the menu tree that was familiar to them, even if you're hoping that they would later switch to your own menu tree?

A. That was not its intention.

Q. When you say that they would perceive it as easy to adopt, why would having a keystroke compatibility with 1-2-3 make it easy for those customers to adopt?

A. It was not a question of actually intent for the customer to use so much as a perception that there was an alternative, what was sold was benefits of the Quattro Pro and its menu and its features.

Q. But to get them to make the purchasing decision, it's an advantage to let them know that if they have doubts about the

Quattro Pro menu tree, they will be getting the 1-2-3 menu tree too.

MS. OLSON: Objection as to form. You can answer.
THE WITNESS: Could you repeat what he said.

(The record was read as follows:

* * *

[56] THE WITNESS: Yes, we provided so that if they—they would perceive that if they—that they had some safe ground that they were familiar with, that the product would be familiar to them.

BY MR. KONRAD:

Q. Insofar as they were already familiar with the 1-2-3 menu tree, that's what would give them some safe ground?

A. There's more to it then that in that.

Q. Is that at least part of it?

A. That for some customers that would be true.

Q. And what else—I don't mean to cut you off, but what else would give them that safe ground?

A. That if the perception that if there was something familiar to them in the product, that the rest of the product would also be familiar, that it wouldn't be a **[57]** completely foreign purchase.

* * *

[58] BY MR. KONRAD:

Q. How about from before then? Do you have any understanding?

A. Nothing specific, just impression.

Q. Your impression being?

A. Very, very, few.

Q. Well, in your view, would it have a negative effect on the marketing of Quattro Pro if you just took the 1-2-3 menus out?

A. Not significantly.

Q. It would have some effect?

A. I really don't know.

Q. In your view has the inclusion of the 1-2-3 menus in Quattro Pro had a positive effect on its sales?

A. I don't know if its had any effect whatsoever.

Q. Have you ever seen any studies or research or attempts to evaluate what effect the 1-2-3 menus might have had on sales of Quattro Pro?

A. I don't recall any specific studies or research.

Q. Now is it fair to say that your views on the point are based largely on the impressions you gathered

* * *

[171] issue specifically from any particular customer.

BY MR. KONRAD:

Q. If you just look at the Getting Started Manual for 2.0 page 38, I'll point you right to it. It says:

> "The Quattro Pro menu tree incorporates menu improvements over the Lotus 1-2-3 version 2.1 menus, but the 1-2-3 compatible menu tree is available for those who would rather use commands that they're already familiar with."

Are you aware that this was said in the manual?

A. Actually I was not aware of those particular words in the manual.

Q. Are you aware that that was ever a point made in marketing Quattro Pro that the commands that were familiar to users were there in the 1-2-3 menu tree?

A. That last statement, I believe to be correct.

Q. That point was made by Borland in selling Quattro Pro?

A. Well, you asked, I believe—could you repeat the question that he did ask?

MR. KONRAD: Yes.

(Question read.)

BY MR. KONRAD:

Q. The commands that were familiar to users were there in the 1-2-3 menu tree, that's the only way it made **[172]** sense,

commands that were familiar to users were there in 1-2-3 menu tree. You said I believe that statement would be correct?

A. I believe that statement is correct.

Q. And forgive me because it has been a long day, and you have been most helpful, but I need to understand that answer better. You mean that Borland did make that point in selling Quattro Pro?

A. I don't know if that was made consistently in everything that was done. In some places, it was noted that that was in fact the case.

Q. In your view was that a benefit in marketing Quattro Pro?

A. As we discussed earlier that it was based, it was used as a means to get customers, potential customers to consider purchasing the product.

Q. Fair enough. I am not going to mark it, I'm just going to show you Bates stamped 509452 through 509457. Just ask you if you have seen it before, if you haven't we'll skip it.

A. (Witness reviews document.)

Q. If you have, we'll mark it.

A. I don't remember if I specifically saw this one.

Q. Well, I'm trying to save time, but that gets

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Volume I
Pages 1 to 166
Exhibits See Index

Civil Action
No. 87-0076-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

PAPERBACK SOFTWARE INTERNATIONAL, et al.,

Defendants.

Civil Action
No. 87-0074-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

MOSAIC SOFTWARE, INC.,

Defendant.

THIS TRANSCRIPT IS DEEMED CONFIDENTIAL
PURSUANT TO COURT ORDER

DEPOSITION OF JOHN POSNER, a witness called on behalf of the Defendant Paperback Software International, taken pursuant to the Federal Rules of Civil Procedure, before Daniel

P. Wolfe, Registered Professional Reporter and Notary Public in and for the Commonwealth of Massachusetts, at Offices of Sherin and Lodgen, 100 Summer Street, Boston, Massachusetts, on Tuesday, March 7, 1989, commencing at 11:00 a.m.

* * *

* * *

[5] BY MR. GUPTA:

Q. Please state your name.
A. John Posner.
Q. Where do you live?
A. 89 Highland Circle, Wayland, Massachusetts.
Q. Are you employed?
A. Yes.
Q. Where are you employed?
A. Stellar Computer in Newton, Massachusetts.

* * *

[7] Q. Would you please outline for us the positions you held upon completing your education and up to the point of your present job.

A. Immediately after I got my Master's of Education, I took a job in the math textbook department at Houghton, Mifflin Company in Boston. I stayed there from June '76 until November of '77. At that point, I got a job as a technical writer at Data General Corporation in Westborough, and so I was employed there from November '77 to February 1981. At that point I took a job with BGS Systems.

Q. Could you say that again.

A. BGS Systems in Waltham, and I was employed [8] there from February '81 through the end of 1981. At that point I resigned and became a contract technical writer and I did a few contracts during the first half of 1982. And on June 1st or—I believe it was June 1st. It perhaps was May 1st but I am fairly certain it was June 1, 1982, I started working as a contract technical writer at Lotus Development.

Q. Did there come a time when you left Lotus?

A. I left Lotus in April of 1986 and joined Stellar Computer and still work there.

* * *

[9] Q. Is it correct, then, that you wrote documentation rather than code as a technical [10] writer?

MR. KONRAD: I object to the form. If you understand it, go ahead.

A. Most of my job was producing written documentation. Part of my job was producing electronic documentation, for instance, help screens, and in particular the Interactive Tutorial for 1-2-3. In the case of electronic documentation, it is—there is a fuzzy line in between writing code and writing documentation, but it would be fair to say that I write documentation, that I do not write code.

* * *

[39] Q. Did you write any documentation for TRIO during the period of time that the project was known as TRIO?

A. Yes.

Q. What was that documentation?

A. I wrote—well, I am not exactly sure when the project was renamed from TRIO to 1-2-3 so that will affect my answer. I will answer it this way: During the summer and fall of 1982, I wrote substantial parts of the written documentation that eventually came out under the name *1-2-3 User's Manual*. I also wrote the scripts for the On-line Help and for the Interactive Tutorial that were eventually delivered with the program 1-2-3.

* * *

[89] Did you generate any documents with respect to designing the menu structure of Symphony?

A. Yes. I generated documents which showed proposed menu structure. I would circulate that to Mitch and to other people for comments.

Q. In the course of designing the menu structure for Symphony, did you consider the menu structure of 1-2-3 to be a jumping-off point?

MR. KONRAD: I object to the form. If that term has any meaning to you, you can answer it.

A. There were a number of constraints, one of which was that Symphony had not only a spreadsheet but a word processor and a data base manager and communications capability as well. In designing the spreadsheet menu structure, I did not feel constrained to be compatible with 1-2-3's menu structure.

* * *

[95] Q. Do you recall the substance of your conversation with Mr. Kapor with regard to the Symphony menu structure?

A. No. He would agree with some things I did, disagree with others. I would make the changes. It was just a boss reviewing a subordinate's work.

Q. Do you recall making any comments to him with respect to functionality in connection with putting names on features?

A. Yes.

Q. What do you recall?

A. I recall with great admiration, after having tried to figure out a good name for the function that would split a window into two windows so that they didn't overlap, after having thought about it for a few hours and not coming up with something, I brought it to Mitch and he said "Windowpane" in about five seconds, and that's what it was.

* * *

[96] In designing the menu structure for Symphony, did you use at any time any document or other recording containing the menu structure for 1-2-3?

A. I am sure I consulted 1-2-3, yes.

Q. In the case of each decision you made in connection with designing the menu structure for Symphony, did you feel free to make a change with respect to a comparable word in the 1-2-3 menu structure?

MR. KONRAD: I object to the form. And I think it is a little unfair to ask somebody with **[97]** respect to each decision made. If you are comfortable answering that generally, I suppose you can try.

A. In general, I did feel that I could change the names of menu items and essentially the structure of the menu.

* * *

Q. I will try it again. Where you did not make a change with respect to Lotus 1-2-3, was that decision not to make a change the result of work independent of 1-2-3?

[98] A. I am not sure what the question is. There was no case that I can recall where Mitch or some other person said "You can call this function AAA because it is called BBB in 1-2-3." I can't recall any case like that.

Q. Did you have a process for making the decision in connection with designing the menu structure for Symphony?

A. Just an untuitive process.

Q. What do you mean by "an untuitive process"?

A. I mean there are no a priori principles that I apply that I learned in graduate school or anything like that. There were certain constraints placed on the choices. For instance, the total number of characters can't be too great because you only have 80 characters across the screen. Each menu choice has to begin with a different letter of the alphabet. So there are various built-in constraints like that.

Q. What other constraints did you have at that time?

A. Only the pedagogical constraints of trying to be as clear as possible.

Q. Were those constraints recorded in a **[99]** document on a media other than paper?

A. Not that I recall, no.

Q. Did you discuss those constraints with anyone?

A. Not per se. They are just part of the business of designing a menu structure, that everybody agrees that you have to live within those—well, certainly the mandated constraints of length and uniqueness of the first letter, that's just a constraint of the program. I am sure everybody would agree with

motherhood and apple pie and clarity as being desirable. How you achieve that is not always a matter of unanimity.

Q .What do you mean by "clarity" in the context of designing the menu structure for Symphony?

MR. KONRAD: I think he told you he described the constraints of trying to be as clear as possible. "Clarity," as I understand it, is the noun form of "clear," but you can explain to Mr. Gupta.

A. When a word appears on the screen, you hope that 99 percent of the people understand immediately what that word means.

[100] Q. In terms of activating a particular command?

A. That's right. So for instance, when I mentioned before Windowpane, if you saw the command "Windowpane," you may or may not immediately be reminded of the process of dividing one window into two equal-sized windows that don't overlap. So that may or may not have been the right choice. It turned out to be the only choice that we could come up with that satisfied the constraints. For instance, I would have chosen "Window Split," but the letter "S" had already been assigned to another menu choice on that same menu, so the letter "S" was not available, so we had to find a word that didn't begin with "S." I had a hard time doing it. Mitch selected the word "pane." No one could come up with something better, so we went with it.

* * *

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

Volume I
Pages 1 to 185
Exhibits See Index

Civil Action
No 87-0076-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—VS.—

PAPERBACK SOFTWARE INTERNATIONAL, et al.,

Defendants.

Civil Action
No 87-0074-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—VS.—

MOSAIC SOFTWARE, INC.,

Defendant.

DEPOSITION OF JONATHAN M. SACHS, a witness called on behalf of the Defendant Paperback Software International, taken pursuant to the Federal Rules of Civil Procedure, before Daniel P. Wolfe, Registered Professional Reporter and Notary

Public in and for the Commonwealth of Massachusetts, at the Offices of Sherin and Lodgen, 100 Summer Street, Boston, Massachusetts, on Tuesday, November 29, 1988, commencing at 10:25 a.m.

PRESENT:

O'SULLIVAN, GRAEV & KARABELL
(by Henry B. Gutman, Esq)
30 Rockefeller Plaza, New York, NY 10112,
for the Plaintiff.

* * *

[154] Q. Then I will snow you Exhibit No. 74. If you would begin at Page 1 and read those commands and point to the ones that fit the description that you and I were just discussing.

MR. GUTMAN: I am sorry. Which description is this?

A.Commands whose name is—

Q. Which were chosen for the same reason that you chose "Insert" and "Delete."

MR. GUTMAN: Because they provide information concerning the function? Is that the question?

Q. Do you understand the question, Mr. Sachs?

A. I think so. I would have to say every command was chosen because it suggested to some measure what the command did. The trade-off in choosing these was to pick the most informative word subject to the constraint that two words in the same menu didn't start with the same letter.

Q. Were these words the most natural words, as I believe you testified earlier, to be selected?

MR. GUTMAN: I object to the form of the question. I don't know what "most natural" means.

A. "Extract" beginning with an "X" is not

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—VS.—

BORLAND INTERNATIONAL, INC.,

Defendant.

The deposition of DOROTHY SCHRADER was taken on Friday, September 13, 1991, commencing at 10:00 a.m., at the Library of Congress, Madison Building, 101 Independence Avenue, S.E., Room LM407, Washington, D.C., before Susanne Q. Tate, Registered Professional Reporter and Notary Public.

* * *

[24] Q. I would now like to have marked as Schrader Deposition Exhibit Number 7 a true and correct copy of a two-page letter from Phill, P-H-I-L-L, middle initial L, Gill to Stuart, middle initial P, Meyer, dated June 15, 1989, and for purposes of clarification, on this letter in the margin, I have placed a mark with a pink felt-tip pen.

(Schrader Deposition Exhibit Number 7 was marked for identifi ation.)

BY MR. REBACK:

Q. The reporter is now placing Exhibit 7 before the witness.

Ms. Schrader, can you identify this document, Exhibit 7, as a true and correct copy of a letter from Phill Gill to Stuart Meyer sent by the Copyright Office on or about the date indicated?

A. Yes.

MR. GUTMAN: Before you go on, I'd like a moment to read it, Mr. Reback.

[25] MR. REBACK: Please. May I continue, Mr. Gutman?

MR. GUTMAN: Certainly.

BY MR. REBACK:

Q. Ms. Schrader, I call your attention to the text next to the pink mark in the margin. Do you see that, ma'am?

A. Yes.

Q. Is this the only letter that the Copyright Office has ever written with that language in it?

A. No.

Q. Is the paragraph marked with the pink mark, which for purposes of identification is the last paragraph starting on the first page and continuing on the second page, is that paragraph language that has been approved by the Copyright Office for inclusion in correspondence from examiners to applicants?

MR. GUTMAN: Object to the form.

MR. REBACK: You may answer.

MR. FARGO: Do you understand the [26] question?

THE WITNESS: Yes.

BY MR. REBACK:

Q. Can you tell us or give us your best approximation of the number of letters sent by the Copyright Office over the years that have that language in it?

MR. FARGO: I'll advise the witness she need not speculate. If she can answer the question, okay.

MR. GUTMAN: I'll object to the form.

THE WITNESS: The paragraph is a standard paragraph that is frequently included in correspondence cases relating to reg-

istration of computer screen displays. The examiner, of course, first examines the claim and determines whether this kind of comment is appropriate, but the paragraph itself is a standard paragraph that has been used many times in letters. I could really quantify the number of times.

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Case No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF ROBERT W. WARFIELD
Wednesday, May 15, 1991
VOLUME I

SPECIALLY CONFIDENTIAL

Reported By: Deborah A. Fryer, CSR #5191

---

* * *

[6] MR. KONRAD: Q. Mr. Warfield when were you born?

A. January 22, 1961.

Q. And are you in good health, sir?

A. As far as I know.

Q. Okay. Is there any physical or mental reason that would prevent you from testifying to the best of your recollection today?

A. No.

Q. Thank you, sir. By whom are you employed?

A. Borland International.

Q. For how long have you been employed by Borland?

A. Almost three years.

Q. Do you recall approximately when you started with Borland?

A. Would have been two years ago last November.

Q. November 1988?

A. Uh-huh.

Q. And what is your current position with Borland?

A. I am the Vice President/Manager of the [7] Spreadsheet and Analytical Tool Development Unit.

MR. KONRAD: Can I just hear that back.

(Record read.)

MR. KONRAD: Q. Could you tell me generally what your duties involve in that position?

A. I'm responsible for the development of spreadsheets and analytical tools.

Q. Are analytical tools, within the terminology of Borland, something different from spreadsheets?

A. In some cases.

Q. Maybe the easiest way to try to understand that is simply to ask you what published products or commercial products fall within your purview?

A. Quattro Pro.

Q. Any others at this time?

A. SideKick.

Q. Any others at this time?

A. No.

Q. And are there any products currently in development, but have not yet been released, which fall under your area of responsibility?

A. Yes.

* * *

[42] Q. Let's refer specifically to Quattro Pro 1.0 or what we're calling "1.0" for now; in your own words, could you describe to me what the 123.MU file is?

A. It's a program that controls the contents of the menus and some of the messages in the product and some other miscellaneous settings and parameters.

MR. KONRAD: Could you I hear that [43] back.

(Record read.)

MR. KONRAD: Q. When you refer to it as a "program," do you mean to say that it is a program distinct from some other parts of the Quattro Pro product?

A. Yes.

Q. And when you use the term "program," do you mean that it was written in a programming language?

A. Yes.

Q. What is that?

A. There is no name that I know of for it.

Q. How would you describe that language?

A. It's the language we use to generate our .MU files.

Q. Was this some language that was developed by Borland?

A. Not entirely.

Q. Was it also used in connection with Surpass?

A. Yes.

Q. Anywhere else that you're aware of?

A. Yes, it was also used in connection with Farsight.

Q. What was Farsight?

[44] A. Farsight was a spreadsheet product I worked on before Surpass.

Q. Who did the actual programming for the 123.MU file in the first version of the Quattro Pro?

A. Depends on which part of the file you're referring to.

Q. What different parts are you referring to?

A. The contents of the menus, some of the messages and the settings I referred to.

Q. So what you're saying is the answer would differ with respect to those different items?

A. Correct.

Q. Well, let's establish more clearly what those different things are and then I'll double-back.

When you refer to the contents of the menus, what specifically are you referring to?

A. The overall system that the commands refer to by the menus are organized in, the words that appear on the menus, the status information that appears on the menus, the long prompts for the menus, and some of the prompting details for the commands.

MR. KONRAD: Could I hear that back, please.

(Record read.)

MR. KONRAD: Q. Could you [45] explain a little bit more what you mean by the "overall system that the commands refer to by the menus are organized in" is what she just said, can you explain exactly what you're referring to there?

A. When you use the menus you trigger certain commands, the menus are linked together under this organizational system and that's the system I'm referring to.

Q. Is a term to use for that the "menu tree"?

MR. GREGORIAN: If you know or understand that term.

THE WITNESS: So you're saying "menu tree" would refer to what?

MR. KONRAD: Q. The overall organization of the menu?

A. You could refer to it that way.

Q. As much to save words, I think I will, unless you object.

When you say the words that appear in the menus, what specifically are you talking about there?

A. There's a lot of different words that appear in the menus, there's command names, there is status information, there's shortcut information, there's symbols.

Q. Would you include within that each, well, [46] any particular menu commands such as "Worksheet," would that be a word appearing in the menus?

A. That could be a word appearing in a menu.

Q. Who was actually in charge of that part of the 123.MU file?

[47] A. That would have been me.

* * *

[49] Q. Did you use the technology involved in the Menu Builder feature in order to create the 123.MU file?

A. What do you mean by "technology"?

Q. Well, I'm trying to be more general so that I don't suggest that the finished product itself was used to create the finished product, which would seem absurd. So all I'm trying to get at, though, is whether the technology that underlines that Menu Builder feature was actually something that you used yourself in creating the 123.MU file?

A. No.

Q. Did you actually sit down and write out a program to create that file?

A. I modified a program to create that file.

Q. And what program was that?

A. Well, I guess you'd call it the 123.MU program, it doesn't have a special name.

Q. Was that something that—let me back up. Was that something that was carried forward from Quattro?

A. No.

Q. At the time that you started to work on it, **[50]** there was already a version of the 123.MU program, is that what you're saying?

A. Yes.

Q. Do you recall approximately when you did start to work on that?

A. Well, there would have been minor changes made all during the course of development.

Q. I guess I'm looking for an initial point, if there's any way that differs from the time that you began, let's say, as a formal employee at Borland?

A. I see. Would have been in 1983, 1984.

Q. I think I'm beginning to see the light. Was the modification that you were doing to a program that you had previously worked on?

A. Yes.

Q. And that was something that had come to Borland at least directly from Surpass?

A. Yes.

Q. Had you originally written that program?

A. Yes.

Q. And in '83 or '84 was that back at Farsight?

A. That company was called Interface Technologies.

[51] Q. Which was the company that published Farsight?

A. Yes.

Q. When you did modify that program while at Borland, can you recall generally what the nature of the modifications were that you had to do?

MR. GREGORIAN: Are we talking about modifications for Quattro Pro version 1.0?

MR. KONRAD: Yes, I mean to talk, we're now at Borland, we're before Quattro Pro 1.0 is out.

MR. GREGORIAN: We've got a program that's being modified?

MR. KONRAD: Mr. Warfield said he made modifications to an earlier program he had done.

Q. And all I want to find out now is can you remember generally the nature of the modifications you made at that point?

A. Well, in general they would have been either due to changes to the syntax of that language that we made, or involving the addition of the commands that were new and unique to Quattro Pro.

Q. And when you refer to the changes in syntax of that language, you mean the programming language without a formal name?

[52] A. Correct

Q. That we talked about before?

A. Yes.

Q. Thank you. When you talk about the addition of commands unique to Quattro Pro, what are you distinguishing those commands from?

A. Commands that were in Surpass.

Q. Now, how long did you work at Surpass?

A. However long they existed, I think it was like two years, maybe a little more.

Q. Approximately 1986 through 1988?

A. Something like that, yeah.

Q. And before Surpass where did you work?

A. Interface Technologies.

Q. And approximately how long were you there?

A. About three years, maybe a little less.

Q. What was your position at Interface Technologies?

A. I held several positions.

Q. Were you a designer of the Farsight product?

A. Yes.

Q. Were you the principal designer of it?

A. What do you mean by a "principal designer"?

Q. Would you say you were in charge of its **[53]** development?

A. Yes.

Q. Were you the designer of its user interface?

A. Yes.

Q. The program that we have been referring to that ended up being modified to constitute the 123.MU file, how did that get from Interface to Surpass?

A. On diskette.

Q. Was it sold, was it your property, I'm just trying to —

A. Oh I see. It was sold to Surpass.

Q. I apologize if you think I ought to know this, but I don't; did Interface Technologies sell its assets to Surpass at some point?

A. Did it sell its assets? What do you mean by "assets"?

Q. Did it sell its programs, did it sell the Farsight program to Surpass?

A. Surpass acquired a technology license for Farsight.

Q. Thank you. Interface Technologies had been located where?

A. Houston, Texas.

* * *

[54] MR. KONRAD: Q. Was the Farsight technology used in the development of Surpass?

A. Yes.

Q. Were you in charge of the development of the Surpass product?

A. Yes.

[55] Q. Were you responsible for the user interface design of the Surpass product?

A. Yes.

Q. Is it fair to say that the Surpass product was an evolution of the Farsight product?

MR. GREGORIAN: Object to the form, it's a little unclear I think. If you have an understanding of what he means, answer the question, but make clear what your understanding is so we don't have a miscommunication.

THE WITNESS: Let's ask, what do you mean by "evolution"?

MR. KONRAD: Q. Obviously you're in a different company, but you'd worked on Farsight, you're now working on Surpass, the technology is being used, and all I'm trying to get at is whether or not the Surpass product was built upon the Farsight product.

MR. GREGORIAN: Farsight technology?

MR. KONRAD: Q. Technology, to be more specific.

A. We used some of the technology from Farsight in the construction of Surpass.

Q. Did you redesign the user interface for **[56]** Surpass?

A. Some aspects of it.

Q. Which do you recall?

A. Primarily Farsight used to start up in a file manager and Surpass started up in a spreadsheet.

Q. Now, was the particular program, the one that ended up ultimately as the 123.MU file, used in the Surpass product too?

A. Yes.

Q. And that had originally been used in the Farsight product?

A. Yes.

* * *

[58] Q. What was the goal in your mind for the menu system of Farsight?

A. There were a lot of goals, we wanted it to be easy to use and easy to learn, and we wanted it to be capable of executing as many Lotus-compatible macros as it could.

Q. And that last goal of allowing it to **[59]** execute as many Lotus-compatible macros as it could, what effect did that have on the design of the menu tree for Farsight?

A. We had to design the menus so they would behave exactly as required to satisfy that compatibility.

Q. And what does that mean?

A. If a macro executes the same keystrokes under 1-2-3 Version 1-A and Farsight, it should trigger the same command.

Q. Did they not mean that the menus of Farsight would have to provide at least the first letter for each command word identical to that from 1-2-3?

A. At least the first letter, yes.

Q. And yet in fact Farsight displayed the full word from 1-2-3, not just the first letter; correct?

A. There were full words displayed in the menus.

* * *

[60] 1-2-3, were also the same as in 1-2-3; correct?

A. I believe in many cases, I'm not sure about all.

Q. In most, it's fair to say, is it not?

A. More than half.

Q. Do you recall, when you designing the interface, making an objective of finding different words that began with the same first letter?

A. No.

Q. Can you recall any particular instance where you chose a word that used the same first letter but was otherwise different from 1-2-3?

A. I believe "Worksheet," for example, may have been "WKST."

Q. Any others that you can think of?

A. Not that I recall.

Q. And in order to achieve as great a degree of macro compatibility as possible, those words for the menu commands had to be displayed within each menu in the same order; isn't that correct?

A. In the same order?

Q. Within a given menu, so that it would go Worksheet, Range, Copy, Move, as opposed to Range, Copy, Worksheet, Move.

* * *

[64] Q. And with respect to the overall menu tree, with respect to those commands that were the same in Farsight as in 1-2-3, they were placed in the same groupings of menus in the same hierarchical organization, again to preserve macro compatibility; isn't that right?

A. Correct.

Q. And that portion of the program stayed the same in Farsight and Surpass and in Quattro Pro?

A. No.

Q. What changes occurred to it?

A. In Surpass we added macro compatibility with Lotus Version 2.01.

Q. And what did that involve your doing?

A. We just added some new picks to the menus.

Q. Some new what?

A. Picks, commands to the menus.

Q. When you used "pick" there you meant "p-i-c-k-s," as in new choices?

A. Correct.

(Discussion off the record.)

MR. KONRAD: Q. And those new picks or choices were commands that were new to Release 2.01 of 1-2-3; correct?

A. That's correct.

[65] Q. And you put them in the same order within the Surpass menus that they occurred in 1-2-3 Release 2; right?

A. Yes.

Q. How about with respect to Quattro Pro, did you make any changes to that part of the program when it became Quattro Pro?

A. I don't recall any.

Q. Now going back to the time that you worked on Farsight, and designing, originally designing the menu tree that ended up in Quattro Pro, how did you make sure that you had the right first letters and had the commands in the right order?

A. A lot of different ways.

Q. How?

A. We executed Lotus-compatible macros on the product, we looked at how macros were executed in Lotus.

Q. What do you mean by that?

A. I mean we executed them and observed the result that happened within Lotus as a user. We read the documentation for Lotus. Here, I keep saying "we," it's me; managers are used to saying "we."

Q. Fair enough, thank you.

A. Read books on Lotus, like by the Cobb **[66]** Group.

Q. When you say—anything more?

A. No.

Q. When you say "looked at how the macros executed in Lotus," you mean specifically you looked at it in operation on the screen, you don't mean you tried to get inside that?

A. We looked at it in operation on the screen.

Q. Okay. And that was specifically Lotus 1-2-3 Release 1-A at that point?

A. Yes.

Q. Regular package of it and just ran it?

A. Yes.

Q. In this program that began with Farsight, ended up in Quattro Pro, somewhere in the lines of code does the text for

the words that appear in the command menus exist actually typed out?

A. Yes.

Q. Somewhere in the code of this program there you could find the text "Worksheet"?

A. Yes.

Q. And the word "Range"?

A. Yes.

Q. And so forth through all the different commands?

[67] A. Yes.

Q. Did you actually type that in?

A. Yes.

Q. What were you looking at when you typed that in?

A. Generally I was looking at the program on the computer and a book.

Q. The program being 1-2-3 Release 1-A?

A. No, no, the program being this program that has no name that wound up being 123.MU.

Q. And the book—I understand. So that's just the programming code right there that you're working on and looking at and typing into?

A. Right.

Q. And the book is what?

A. The book would be like this one I referred to by Cobb, those kinds of things.

Q. How about the menu tree in the back of the 1-2-3 manual, did you look at that?

A. I didn't know there was a menu tree in the back of the 1-2-3 manual.

Q. Is it your recollection that at someplace in the Cobb book there is a listing of all the different 1-2-3 menu commands?

A. Don't recall.

[68] Q. Can you recall going back to 1-2-3 Release 1–A and verifying that you'd gotten the words right and in the right order?

A. Not to the program.

Q. To the documentation?

A. Yes.

Q. Any particular place in the 1-2-3 documentation that you can recall?

A. As I recall, the documentation has the commands scattered all through the manuals, so it would have been that sort of a process.

Q. And when you were typing those words in, whether you were looking at the Cobb book or some other book or the 1-2-3 documentation —

MR. GREGORIAN: Object to the form.

MR. KONRAD: Q. Well, the thrust of my question is not what you're looking at at the time.

MR. GREGORIAN: Okay. The way you prefaced it, it can be misconstrued and I'm sure that isn't your intent.

MR. KONRAD: As you know, it is not my intent ever.

MR. GREGORIAN: Okay, we're on the same wavelength then, we want a clear record and **[69]** that's why I objected.

MR. KONRAD: Fair enough.

Q. When you were typing in those words, regardless of whatever you may have been using to assist you in that, the intent was that the words that would be displayed in the Farsight menus would be the same as those displayed in 1-2-3 Release 1-A; is that not correct?

A. No, the intent was to achieve the macro compatibility.

Q. But you understood that the result of what you were doing typing in those words would be to cause Farsight to display the same words as those displayed in 1-2-3, didn't you?

A. Yes.

Q. Now, again, to be very specific, and Mr. Gregorian in fact does me courtesy because he knows I am just simply trying to be clear for the record, can you recall when it came time to complete that program for Quattro Pro making changes to that portion, the words, the menu setup portion of that program?

A. Sure.

Q. What changes did you make?

A. We added the new commands and stuff that we talked about.

* * *

[76] Q. When you were working on Farsight, it's fair to say, isn't it, that displaying the words in macro debugging was not part of your concern since the product didn't have that ability; right?

A. No.

Q. I'm wrong?

A. You're wrong.

Q. Okay, why?

A. While we didn't have a macro debugger per se, there was a single stepping capability and there was certainly a necessity for the user to be able to debug their macros.

Q. When you say that when macros are being executed they will display the menus, I'm sure I didn't get it all down in notes, but what did you mean by that?

A. I mean it's possible that as part of the normal functioning of a macro it may display a menu that the user is supposed to make a choice from.

Q. Why do you need the full word to be displayed at that point any more then when you're not running a macro?

A. So you have some clue what the menu means.

Q. In other words, the words give you a clue as to what the choices are?

[77] A. Yes.

Q. It's because those words are conveying some information to the user about the nature of the choices the menu presents; correct?

A. Words have meaning.

Q. And the words that are displayed there have some meaning that tells you about the choices you can make at that point; correct?

A. Combination of the words and long prompts.

Q. And when it comes to debugging macros, having the full word displayed does the same thing, does it not, it gives some —

MR. GREGORIAN: What is the "same thing"?

MR. KONRAD: Q. It gives some information to the user about what the various command choices are?

A. As well as some information about what the macro is actually doing.

Q. Does it give information to the user?

A. Yes.

Q. And that's because a full word provides more information than a single letter; correct?

A. That's correct.

Q. And what did you mean, I don't mean to

* * *

[78] Q. Isn't it true, though, that simply, if your objective was simply to provide the ability to execute macros previously written in 1-2-3, you wouldn't need to display the full word?

A. No.

Q. Why not?

A. Part of the process of executing a macro is that a user has to be able to successfully use the macro and make choices from these menus.

Q. Well, do you have any sense as to how many macros are written in such a way that they pause and ask the user to make some choice in the middle of its execution?

A. It's not an uncommon practice.

Q. Do you have any estimate in your mind as to how many, as to what proportion of macros are written using that?

A. No.

[79] Q. Have you ever seen any information in written form about that?

MR. GREGORIAN: About the percentage?

MR. KONRAD: Uh-huh.

THE WITNESS: No.

MR. KONRAD: Q. Have you ever heard that anyone at Borland has made any effort to try to quantify what proportion of macros actually use this practice of pausing to permit a user to make some choice?

A. No.

* * *

[124] Q. How did the goal of making the product easy to learn affect the design of the user interface for Farsight?

A. It would be similar to ease of use, although somewhat extended. We would have included emphasis on the "Help" facility, and emphasis on the documentation of the product, those kinds of learning materials.

Q. Would it have affected the choice of menus and menu commands?

A. What do you mean by "would it have affected the choice of menus and menu commands"?

Q. Did the goal of making the product easy to learn affect the choice of menus and menu commands?

A. Yeah.

Q. How so?

A. Obviously we had to choose some menus that would make some sense to the user, and commands had to have terms that were, you know, understandable or familiar, usual and customary, that kind of thing.

Q. Is it fair to say that the choice of command terms and the arrangement of the menus plays a part in the user's learning of a new product?

A. Yes.

* * *

[126] Q. Did you think that was within your capability at the time?

A. I don't recall having thought about it in that degree of depth.

Q. Just to try to wrap this up, at no point did you ever really try to construct a menu tree for Farsight that was noncompatible with 1-2-3?

A. Correct.

Q. Do you recall whether at that time back working on Farsight you formed any opinion as to whether the 1-2-3 menus and menu tree was the best possible menus that you could have for a spreadsheet product?

MR. GREGORIAN: You're asking for his personal opinion?

MR. KONRAD: Yes, I'm strictly asking for whether when he thought about it he formed any opinion on that idea.

MR. GREGORIAN: First did you form an opinion on that, and then his next question will be what was it.

THE WITNESS: Well, the question of what's the best possible menu tree is a difficult one, what's the best possible with respect to what?

[127] MR. KONRAD: Q. What would you want to know in order to answer that?

A. Well, I'd talked about three objectives, ease of use, ease of learning and macro compatibility; "best possible" could be different in each of those cases, for example.

Q. Well, do you recall back then at least whether you thought that if you design a menu tree from scratch, one that was fitted particularly to Farsight and what that product was going to have, you could do a better job than the 1-2-3 menu tree?

A. With respect to which one of those three goals?

Q. With respect to ease of use and ease of learning let's say.

A. Yeah, it occurred to me a better job could be done.

Q. I take it that the third objective of compatibility overrode those other two; isn't that right?

A. To some extent.

Q. I mean they are inconsistent, are they not, to some extent?

A. To some extent.

Q. Is it possible to, I mean I guess I'm **[128]** trying to, to me there's a logical problem, you can't be macro compatible, according to what your testimony is, and provide a completely different menu tree, isn't that right?

A. Except in the sense that Quattro Pro is capable of providing a completely different menu tree.

Q. When you say "completely different menu tree," let's make sure this is clear; are you referring to the 123.MU file or are you referring to the Quattro.MU file?

A. The Quattro.MU file.

Q. Which in your opinion is a completely different menu tree from the 123.MU file?

A. Yes.

Q. Did you work on that too?

A. Yes.

Q. Would you call yourself the designer of its user interface?

A. One of the designers.

Q. Fair enough. Would you call it a collaborative process?

A. Yes.

Q. Who were your colleagues?

A. Rob Dickerson.

* * *

[150] Q. In designing the Quattro.MU menu tree for Quattro Pro, did you and Mr. Dickerson consider yourselves to have a clean slate?

A. What do you mean by a "clean slate"?

Q. Were you free to choose the menu commands that you thought were the most appropriate and organize them in the menu tree that you thought was most appropriate for Quattro Pro's functionality?

A. We were free to decide what the ground rules would be for determining that organization.

Q. Intersting answer. Could you explain what **[151]** you mean by the "ground rules for determining that organization"?

A. Well, we used some rules when we figured out how to lay out the menus.

Q. Can you give me—can you recall those rules?

A. Oh no, not all of them, there were a bunch of different rules.

Q. Which ones can you recall now?

A. We were seeking to create menus that would be to some degree compatible with the CUA standard, we wanted to build menus that were consistently organized, meaning similar functions were grouped together.

We wanted our menus to require as few keystrokes as possible, meaning they would be very flat. We wanted the more

common commands to be more easily accessed near the top of the menu tree.

I had another one, but it's gone. There's a lot of different rules involved in UI design, a "seven plus or minus two" rule, we wanted menus to have on average seven, plus or minus two choices, on them.

* * *

[154] Q. In terms of choosing the words to represent various commands in the menu tree for the Quattro.MU file, did you and Mr. Dickerson feel yourselves constrained to pick words that were in the first version of Quattro, for example?

[155] A. Not especially.

Q. Did you feel constrained to pick words that were in the 123.MU file?

A. Not especially.

* * *

[159] MR. KONRAD: Q. Yes, that's the problem.

(Record read.)

THE WITNESS: My recollection of that process doesn't extend to that level of detail.

MR. KONRAD: Q. Let's address Mr. Gregorian's point; can you recall any manner in which the 1-2-3 menu tree somehow affected the choices you had available to you in designing the Quattro menu tree, other than the proverbial "first letter of the word" problem?

A. Yes.

Q. What were those?

A. Well, in very broad terms, the two menu trees are in essence rather like filing systems for the same information.

Q. And how does that concept find articulation in the Quattro menu tree?

A. Well, it's like you had been handed some random collection of files, i.e. commands, and asked to organize them in some way, obviously the organization will have something to do with the files you've been handed.

Q. In this metaphor, what comprises a di[illegible]reet file?

[160] A. A command.

Q. Well, the command set for Quattro Pro was different from the 1-2-3 command set, wasn't it?

A. Different, what, meaning, what do you mean by "different"?

Q. Well, first of all, it was bigger, wasn't it?

A. It was bigger.

Q. Any idea how much bigger?

A. Well, there's lots of stuff in there that's not in 1-2-3, I don't know how to quantify that.

Q. I'm just asking whether you can. Was it 100 new commands, 200 new commands?

A. I don't know exactly.

Q. Any idea how many files, using your metaphor, how many different commands there were from 1-2-3 that were in the set handed to you?

A. Meaning from 1-2-3 the product or 123.MU?

Q. Well, let's say 1-2-3 the product—200, 300?

A. I would just have to hazard a wild guess.

Q. You really don't recall?

A. No.

* * *

[165] Q. Oh that's right, absolutely. Now, when you were picking the commands, the menu commands for the Quattro.MU and how they should be arranged—let's back up, let's extend the metaphor, if possible.

Does each folder represent a single command, or in some instances is it a sequence of menu commands? Do you understand what I'm saying? Is there a folder that says "W" for "Worksheet" in the main menu, or is that folder actually /WGF fixed?

A. No, the process of designing the menus is rather like if you had a filing cabinet that could only hold however many of these things will fit on a particular menu, however you can put another filing cabinet inside, meaning you can nest down

to another menu and you go through the process again of organizing for that filing cabinet what happens.

Q. Is there a menu equivalent command for what the program does in response to just "Worksheet" on the main menu?

A. No.

Q. Now, trying to go all the way back to your first use of this metaphor, we're trying to get at how the 1-2-3 menu tree affected the Quattro menu tree in some sense, other than just avoiding the **[166]** first letter problem. We've tried to expand on this metaphor so maybe it's a little flush here.

Can you explain now how your image of these things as filing systems meant that your choices in the Quattro menu tree were somehow or other affected by what was in the 1-2-3 menu tree?

A. Well, we knew we wanted to be able to execute all the same commands that a macro might possibly wind up wanting to execute.

Q. A 1-2-3 macro?

A. Correct.

Q. So you wanted to make sure that there was some Quattro command in this case which could be a sequence of different menu items that would perform an analogous function?

A. The same functionality would be there.

Q. But that didn't necessarily have to be two commands or two keystrokes, whereas in 1-2-3 it was two keystrokes, it didn't necessarily have to be the same menu levels; isn't that right?

A. Well, if you mean does Quattro.MU have different numbers of keystrokes to get to these commands, yes.

Q. Right, and you can look at the menu equivalent tables and see in some instances there are **[167]** five in Quattro and four in 1-2-3 and probably more instances it's fewer in Quattro than in 1-2-3; right?

A. Yes.

Q. Now, did you consider how many keystrokes it took in 1-2-3 when you were trying to figure out how deep to nest the menus and how many keystrokes it would take in Quattro?

A. No.

Q. Well, what was the—maybe I'm just blocking, I'm sorry, I'm working very hard to try to understand it. How does this filing system metaphor explain what effect the 1-2-3 menu tree had on the Quattro menu tree?

A. Well, it would be if you took all the files out of the filing cabinet that was filed that way and removed all their labels and set them on the table and then said, "Okay, now, how can we re-file these so that they're easier to get to?"

Q. What would that be comparable to?

A. I'm just saying—what you said is what does this have to do with the 123.MU. I'm saying it's like you took all of those files out of that filing cabinet that had been filed in that particular manner and said—took their labels off, said okay, now let's figure out how to re-file these in a **[168]** different filing cabinet, Quattro.MU.

MR. GREGORIAN: Along with some other files that weren't in the original filing cabinet?

THE WITNESS: Well, I'm speaking of 123.MU.

MR. KONRAD: Q. Right, I understand, I understand. So the set of files that you already have has been determined, at least to that extent?

A. By what was in our 123.MU system.

Q. Right, that comprises a big chunk of the files you have to reorder?

A. Correct.

Q. And that's sort of what you're trying to get at, that defined a large chunk of the functionality you felt Quattro should provide?

A. It was a large chunk of the functionality we had.

Q. Fair enough, fair enough. Let's take a quick break.

(Recess.)

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Case No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.,

Defendant.

DEPOSITION OF ROBERT W. WARFIELD
Thursday, May 16, 1991
VOLUME II

SPECIALLY CONFIDENTIAL

Reported By: Deborah A. Fryer, CSR #5191

---

[194] Q. Was there a period of time between your work for Interface and your work for Surpass?

A. Yes.

Q. How long?

A. As I recall, it was six to nine months.

Q. Trying to stay focused on the period of time that you were at Interface, did you come across any data or research discussing how many Lotus 1-2-3 users used macros?

A. I can remember one piece of research, but I don't remember if it was in that time or not, it could have been.

Q. This one piece of research is the one piece you remember of, one piece of information about how many 1-2-3 users used macros?

A. I'm sorry?

Q. You say you recall one piece of information, you don't know when it's form, all I'm trying to get at is this: Is the one piece of information you remember about how many 1-2-3 users actually used macros?

A. Oh no, I recall I think three different pieces of information.

Q. Okay, let's establish what those are and **[195]** then see where we can place them chronologically.

A. I think I remember having seen two different Lotus Magazine surveys of their user base, and I think one of our internal, meaning at Borland, market researchers inquired about that.

Q. Do you have any recollection as to when these Lotus Magazine surveys came out?

A. I don't.

Q. Do you recall what either of them said on that subject?

A. As I recall, the first one said something like 40 percent of users use macros and I think the second one was a little higher than that.

Q. But it's fair to say that this information was something you didn't have before you made the decision to be macro compatible?

A. That's correct.

Q. And the third piece, the Borland internal market research, you wouldn't have either obviously?

A. Right.

Q. Do you recall approximately when that piece of information reached you?

A. Sometime after Quattro Pro 1 shipped, it may have been just before Quattro Pro 2 shipped or a little bit after.

* * *

[212] Mr. Konrad: Q. Mr. Warfield, it was an objective in the design of Surpass for it to be 100 percent macro compatible with 1-2-3, was it not?

A. And by "100 percent" you mean?

Q. As close to 100 percent as you could make it?

A. That's correct.

Q. And in your opinion, that meant that you not only had to provide each menu command found in 1-2-3, but also display the words for each of those menu commands in the Surpass menus; isn't that right?

A. That's correct.

Q. And it also meant, in your opinion, that you had to display those words in the same arrangement in the menus as in 1-2-3, except where you were putting additions in Surpass; right?

A. From the macro's perspective that's true, yes.

Q. And all of that was true at the time that you heard about the suit involving VP-Planner by Lotus and wondered whether Lotus might sue you too?

A. Yes.

* * *

[279] MR. KONRAD: Q. Who within Borland do you think would be in the best position to judge how important the 123.MU menu tree has been to Quattro Pro's commercial success?

A. Probably the marketing and sales people.

Q. Anyone in particular?

A. Well, of course, I don't know who the best would be, I just know who I know over there.

Q. Fair enough. Turning to the Quattro.MU menu tree for a moment, do you believe that that menu tree is better suited to Quattro Pro than the 123.MU menu tree?

A. Well, earlier we had talked about three criteria for evaluating a menu tree; which one of those three are you interested in looking at?

Q. Let's make sure I understand which of the three criteria are there you're thinking of, can you give those to me again?

[280] A. Ease of use, ease of learning and macro compatibility.

Q. Well, with respect to ease of use, do you think that the Quattro.MU menu tree is better suited to Quattro Pro than the 123.MU menu tree?

A. What does "better suited to Quattro Pro" mean?

Q. Do you think it's a better menu tree for the product?

A. I think it's an easier to use menu tree for the product.

Q. And for ease of learning, do you think the Quattro.MU menu tree is better for the product?

A. Don't have an opinion on ease of learning.

Q. Why not?

A. I've never had any real opportunity to study how easy it is to learn, I wrote it so I knew it a priori.

Q. Well, did you feel when the product was done that you had done a good job in designing that menu tree?

A. Yes.

Q. Are you pleased with the result?

A. Yes.

Q. Do you think that it's a superior menu tree [281] to that from 1-2-3?

A. In the sense that it's easier to use, yes.

Q. If you were to do it again today, that is try to design the Quattro.MU menu tree, are there things you think you could do better this time?

A. Not really.

Q. Did you ever find it awkward trying to place the new Quattro Pro commands into the 123.MU menu tree?

A. No.

Q. Are there any times where you thought "We're really just sticking this in here somewhere but it doesn't really make sense within the context of the 1-2-3 menu tree"?

A. No.

Q. You felt that you found a logical place for each of the new commands within that tree?

A. I thought so.

MR. KONRAD: I'd ask the reporter to mark as Exhibit 15 a document bearing production numbers 518059 through 518065, it's entitled "Status Report on Surpass 2.0."

(Whereupon, The Above-Described Document Was Marked Plaintiff's Exhibit 15 For Identification.)

* * *

[341] Q. And then you respond: "This is essentially correct." Then you give some exceptions.

Was the contemplation at this time in the summer of '88 that there were just going to be these two user interfaces?

A. That's correct.

Q. When did a decision get made to provide a third different user interface, the one that you and Mr. Dickerson designed?

A. Shortly after we completed the design for that user interface.

Q. Well, what led you to go ahead and design that new interface?

A. We thought it had potential to be a better interface and we wanted to go ahead and try and design a menu and see how we liked it.

Q. Did anyone, other than yourself and Mr. Dickerson, conclude that the new one you came up with was better and ought to be implemented?

A. Yes.

Q. Who else?

A. Most people who saw it liked it.

Q. Was it a concern when you decided to do that that you might be confusing people who were familiar with Quattro menus?

[342] A. I didn't have that concern.

Q. Why not?

A. Because I figured if they preferred the Quattro menus they could just use the Quattro menus.

Q. Well, can you recall any consideration as to what the marketing effect might be of changing the default menu tree from Quattro to Quattro Pro?

A. I don't recall any consideration of that.

Q. Can you recall what led you to think that you could create a better menu tree than that in the original Quattro?

A. I think it was just kind of my liking of the CUA style layout of such things, perhaps just coupled with the feeling that it would be an interesting exercise.

Q. Did you just not like the original Quattro menu tree?

MR. GREGORIAN: Object to the form, I think he's told you why he did it. Okay, go ahead and answer.

THE WITNESS: I was not particularly familiar with the original Quattro menu tree.

* * *

[356] Q. Well, do you recall whether you in fact made any efforts in designing the Quattro Pro menu tree to not go out of your way to confuse 1-2-3 users?

A. I made every effort not to confuse anybody when I designed those menus.

Q. Fair enough. Because one of your goals was that those menus be clear and informative to as broad an audience of users as possible; correct?

A. Correct.

Q. But do you recall specifically having a concern in that regard that the different way that these menus were set up, for example, the use of object verb phrasing instead of verb object phrasing, would be particularly difficult for 1-2-3 users to learn?

A. No.

Q. In your mind there was no particular reason that a person familiar with 1-2-3 couldn't adapt and learn the new Quattro Pro menu tree as easily as they could have learned the new Quattro menu tree, I mean the original Quattro menu tree?

A. That's correct.

Q. In fact did you think that it would be possible for a 1-2-3 user to learn the new Quattro

* * *

[364] MR. KONRAD: Q. Do you understand what I mean?

A. Do you mean ease of use or ease of learning?

Q. What I mean is clarity in expressing what different commands meant to the user.

A. Oh, I see. I hadn't ever really thought of the menus as expressing what commands mean.

Q. What do the words do that are used in the menus?

A. They make it easier for the user to remember how to trigger a particular command.

Q. And how do they accomplish that?

A. By typing on the keyboard.

Q. No, how do the words make it easier for the user to remember?

A. Oh, because people, when they remember commands they tend to sound things out in their heads, such as "File Open."

Q. And the words mean something when they [365] appear in the menus, don't they?

A. Yeah.

Q. Would you say it would be equally good to use the word "zebra" instead of the word "file" in a menu tree if the command that that was associated with had something to do with the filing of data?

A. Not from a standpoint of ease of learning, but possibly from a standpoint of ease of use.

Q. And why wouldn't it be good for ease of learning?

A. They wouldn't be able to infer what it did from "zebra."

Q. Because "zebra" would be illogical and senseless in that context, wouldn't it?

A. Illogical and senseless, yeah, I guess you could characterize it that way.

Q. In other words, the words that you pick to represent different commands ought to have some kind of meaning associated with what those commands do, shouldn't they?

A. Yeah.

Q. And that's something you tried to do in creating the Quattro Pro menu tree, isn't it, pick words that had some meaning to users?

A. Yeah.

[366] Q. And you wanted to be as clear in that meaning as you could be, didn't you?

A. Yes.

Q. Now, you've made reference during this deposition to all these books about user interface design, there are courses on computer users interfaces and the like, do you recognize the name Ben Schneiderman in that regard?

A. No.

Q. Never heard of him? Any books you can recall on that?

A. There's a few I recall the titles of, there was one called I think it was called Principals of Interactive Programming Environment, by Barstow. I think there was one on Designing Integrated Software by a fellow by the name of Newman, who's also famous for a graphics book by Newman and Sprowl.

There have been many articles that appeared in journals of various kinds. There's literature associated with the Apple Macintosh and accounts of what took place there and at Xerox Park. I don't remember any more specific than that.

Q. Well, is it your sense of the teaching in that field that good user interface design has nothing to do with aesthetics or subjective

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

Case No. 90-11662-K

LOTUS DEVELOPMENT CORPORATION,
Plaintiff,

—vs.—

BORLAND INTERNATIONAL, INC.
Defendant.

DEPOSITION OF ROBERT WARFIELD

Monday, November 9, 1992

VOLUME II

CONFIDENTIAL—ATTORNEYS EYES ONLY
Reported by: Kimberlee Cricks, CSR #7473

---

* * *

[74] Q. Are you familiar with a feature in Quattro Pro called the Key Reader?

A. Yes.

* * *

[75] Q. Would you describe to me generally what that Key Reader Command is.

A. The Key Reader allows Quattro Pro to be able to execute some macros without displaying menus for the macros.

[76] Q. When you say "some macros," do you mean macros written specifically in any particular macro language?

A. Yes. These are macros written in a 1-2-3 compatible macro language.

Q. When you say "some macros," is there some way for you to identify which macros the Key Reader feature will or will not execute?

A. The Key Reader feature will execute macros that do not specifically call for the display of the menus.

Q. What types of macros call for the display of menus?

A. Macros that contain the question mark command at an appropriate point where the program would have to stop, the macro program, that is, and display a menu for the user to give a choice.

Q. Any others?

A. No. I think that's it.

* * *

[80] Q. You worked on all the DOS versions of the Key Reader?

A. Yes.

Q. Did you work on the WINDOWS version of the Key Reader?

A. Yes.

* * *

[83] Q. Well, whenever the feature was added, initially, who worked on its design?

[84] A. I did.

Q. Anyone else?

A. Dave Anderson.

Q. Anyone else?

A. No, I don't think so.

Q. And, initially, at least, who worked on its coding?

A. Dave Anderson.

Q. Anyone else you recall?

A. No.

Q. Do you recall who determined to include that feature in Quattro Pro?

A. That was my decision.

Q. Did you consult with anyone with respect to that decision?

A. Yes.

Q. With whom?

A. The consultation took place with attorneys present.

MR. DETKIN: You can identify the people who were there.

THE WITNESS: Philippe Kahn, Bob Kohn, Stephen Kahn, and myself.

* * *

[88] all I can really comment on is that, as I mentioned in the beginning, the Key Reader does allow certain macros to be executed without having the 1-2-3 menu trees present.

MR. KONRAD: Q. Well, who had the very first idea for including such a texture in Quattro Pro?

* * *

* * *

Q. Where did the first idea of doing it come from?

A. It came from me.

Q. Do you recall approximately when, in relation to the meeting that you've mentioned with Mr. Kohn and both Mr. Kahns, this idea occurred to you?

A. Sometime before.

Q. Did the idea occur to you after Lotus sued **[89]** Borland?

A. Difficult for me to recall. It could have been very close to the time.

Q. Well, do you recall how the idea came to you, what led you to that idea?

A. Yes. As I said, the primary reason for including the feature was to make it easy for an end user to execute compatible macros without having to change the default menu tree, and the Key Reader was simply an idea that I had to facilitate that.

* * *

[94] Q. To your understanding, what was done to create the Key Reader feature?

MR. DETKIN: From the design standpoint or the coding standpoint?

MR. KONRAD: Well, let's begin with the design standpoint, which may be easier to discuss.

THE WITNESS: So are you asking, then, **[95]** for kind of a capsule summary of how it works?

MR. KONRAD: Q. That would be helpful. We'll see if we need to go further.

A. Basically, what we decided to do to implement the Key Reader was to include a menu structure consisting only of first letters that would be hidden entirely from view by the user. I think we have referred to that in the past as a phantom menu.

During macro execution, if the Key Reader was turned on, the macro interpreter would simply refer to those phantom menus rather than referring to the menus displayed for the user.

Q. Did you review any documents in preparation for this deposition?

A. Yes.

Q. Did you see the one piece of mail referring to the phantom 1-2-3 menu?

MR. DETKIN: I object. You're trying to invade the work product, and I instruct the witness not to answer that question.

MR. KONRAD: Q. Have you ever seen any document, Mr. Warfield, referring to something identified as the, quote, "phantom 1-2-3 **[96]** menu," closed quote?

A. Yes, I have.

Q. Is that what you were referring to in your answer before as the phantom menu, something that would be described as the phantom 1-2-3 menu?

MR. DETKIN: Object to the form of the question.

THE WITNESS: It's a little bit unclear what you're asking me. Could you maybe either state it again or state it differently.

MR. KONRAD: We'll move on.

Q. You said you created a menu structure that included only the first letters. Now those would be the first letters from the words in the 1-2-3 menus, correct?

A. Let me be more precise. It is not necessarily in every case a first letter. What it is, is whichever letter would be required to trigger that particular menu choice.

Q. Can you describe the circumstance in which the letter required to trigger a menu choice would not be the first letter of that menu command.

A. For example, in our default menus, you exit [97] Quattro Pro with /FX, file exit, where X is the second letter of exit.

Q. Did you create a phantom menu structure for the native or default menus?

A. No. That's just an example.

Q. Can you think of any such example in the 1-2-3 compatible menus?

A. Not off the top of my head.

Q. Did Mr. Anderson actually create this phantom menu structure?

A. Yes he did.

Q. Do you have any understanding as to how he did so?

A. I believe he did so by editing the 123.MU file.

Q. Do you have any understanding as to how he did that?

A. He would have extracted just the relevant information, which is to say the necessary pick letters and the order for those letters and the menu equivalents that needed to be executed for each of those letters.

Q. When you say "relevant information," you [98] mean the information that would be necessary in order to allow the execution of macros?

A. Yes.

Q. Would there be any other information, other than what you just mentioned, that would be necessary to allow the execution of macros?

MR. DETKIN: You're talking now about the macros that the Key Reader is capable of executing, I assume.

MR. KONRAD: Precisely.

THE WITNESS: There may be some odds and ends. I don't remember exactly, but I believe it's the pick letter and the ordering and the menu equivalents that correspond to each pick letter.

MR. KONRAD: Q. When you say "pick letter," I think that's p-i-c-k.

A. Correct.

Q. You mean the first letter of the 1-2-3 menu command, correct?

A. Not necessarily. It's the letter that triggers the particular command from the keyboard.

Q. But with respect to any command in the 123.MU file that is in 1-2-3, that's the first letter, [99] correct?

A. That's correct.

Q. And so for all of those commands, the pick letter would be the first letter, correct?

A. Yes.

Q. And are you, personally, aware of any command in the 123.MU file that is not the same as in 1-2-3 which uses something other than the first letter for its invocation?

MR. DETKIN: Object to the form of the question.

THE WITNESS: So you want to know if there's any command in the 123.MU file that is a command not present in 1-2-3 that has something other than the first letter as a pick letter.

Did I understand that correctly?

MR. KONRAD: Q. The example you gave before, I understood what you were saying. In the native or default mode, you used the file exit convention and X as the operative letter. But that is not the way that you exit the 123.MU file. And I'm asking if you can think of any example where you don't use the first letters in 123.MU for any of the new [100] commands in Quattro Pro.

A. I don't know of an example off the top of my head.

Q. That's all I was asking.

And when you say the order that you extracted, the order for those letters, you mean the order in which those commands appear in the 1-2-3 tree, correct?

A. Correct.

Q. Now when you talk about the menu equivalent commands, you're referring to something there that's unique to Quattro Pro, correct?

A. Many of them also appeared in Quattro.

Q. Fair enough. But for the record here, I want to establish just specifically what we're referring to.

In the Quattro products, you have a macro language that's referred to as the menu equivalent commands, correct?

A. Yes.

Q. Those are not menu tree specific, correct?

A. Yes.

And you can write macros using that **[101]** language which will execute regardless of the menu tree file that is active, correct?

A. Yes.

Q. Now you said that your understanding is that Mr. Anderson edited the 123.MU file to create this phantom menu structure.

In the version of Quattro Pro for DOS that you were working on, did the 123.MU file contain all the words from the menu commands?

MR. DETKIN: Object to the form of the question.

THE WITNESS: Can you restate it. I didn't get the question.

MR. KONRAD: Let me try it again.

MR. DETKIN: Kerry, before you do, we've been going for about an hour. Can we take a quick break?

MR. KONRAD: Can we do two minutes so I don't lose this?

MR. DETKIN: Before you answer that, let's go off the record for a second.

(Recess.)

* * *

[102] it your recollection that the Key Reader feature appeared in the first version of Quattro Pro released after Lotus sued Borland?

A. My recollection was that it appeared in 3.0, which, of course, I know that wasn't the first version.

Q. I'm actually trying to just sharpen this without referring to the numbers.

Can you recall today whether it came out in the next version of Quattro Pro after Lotus sued Borland?

A. Boy, I just don't recall.

Q. Fair enough.

Do you actually recall today when Quattro Pro 2.0 came out?

A. It seems like it was something like approaching twelve months after 1.0 was shipped.

Q. That would be around the fall of 1990?

A. Yeah, I think so.

Q. Going back to where we were before the break, when Mr. Anderson edited the 123.MU file to create this phantom menu structure, did he create a new file in this process?

[103] A. Yes.

Q. What is that file called?

A. Oh, the file temporarily existed only. It got merged into QUATTRO.MU.

Q. When you say it temporarily existed, for a period of time, there was some separate file that Mr. Anderson was working on?

A. Yeah, a work file.

Q. Do you know whether that had a name or—

A. No. Don't know.

Q. Do you know whether any copy of that work file still exists?

A. Don't know.

Q. Did you ever have a copy of that file?

A. No.

Q. Did you ever see it yourself?

A. No.

Q. Did Mr. Anderson describe it to you, or are you simply assuming that such a file had to have existed?

A. I'm assuming the file had to exist.

Q. Thank you.

And for clarity sake, why would you assume **[104]** that such a file had to exist temporarily?

A. It just would have been the easiest way to go about editing, getting the desired end result.

Q. Recognizing that you're assuming, what would that easiest way have been to create that file?

A. To extract just the letters and menu equivalents into a separate file and then to merge that file with QUATTRO.MU. We have to get rid of all the irrelevant parts of the 123.MU file.

Q. And I think that leads perhaps better into what I was trying to get at before the break, which is an understanding of what those irrelevant parts of the 123.MU would be.

A. I see.

Q. Can you explain what parts of the 123.MU file would be irrelevant for this purpose.

A. Okay. One big part of it is that the menus themselves are divorced from being any part of any executable code. So, you see, there is no need to carry that along. There's no need to carry along the rest of the menus that would not be displayed, such as long prompts. Obviously, since we're only taking the pick letter and not the entire word, that information

* * *

[112] Within the portion of the menu information section that has the phantom menu structure, do letters from 1-2-3 compatible menu commands such as WGF actually appear in that section?

A. Yes, they do.

Q. And how is the order—the appropriate order of those letters preserved within that section?

A. Well, it's—as we've discussed, it's an outline format. We've talked about this in prior depositions.

Q. Yes.

A. So they just appear in that same kind of outline format where the depth of the outline implies **[113]** menus leading to other menus.

Q. Well, in the deposition that we did by telephone—and I appreciate both of your efforts in that. I thought that worked very, very well. And I appreciated the professionalism that you both had in making that work—but you described the way in which you had originally written the long prompts. You described this outline format.

Is it your understanding that Mr. Anderson simply took that same piece of 123.MU and just deleted the full words?

A. That is my understanding.

Q. And that outline format that you were using at that time, when was that first created?

A. Well, that was a descendent of a file created long, long ago in the Farsight program, and it was carried forward to Surpass in Quattro.

* * *

Q. Now in addition to deleting the information **[114]** unnecessary for running macros, did Mr. Anderson have to do something so that these would not display to the user?

A. Yes. He modified the internal workings of the product so that when they found the separator character we discussed, that it was understood that the menus would not be displayed.

Q. Well, is that some other part of the program that had to be modified other than 123.MU?

A. That's the executable—the Q.EXE program.

Q. And that's the main executable body of the program, correct?

A. Correct.

Q. What exactly happens when somebody selects Key Reader, turns that on?

A. That option tells Quattro Pro to be alert for the slash key within a macro. And if it encounters a slash, it then invokes the Key Reader to tell it after a sequence of letters which menu equivalent command should be executed.

Q. When you say it invokes the Key Reader there, you mean it goes to that section of QUATTRO.MU that contains the edited form of 123.MU that Mr. **[115]** Anderson created?

A. Right. As we discussed, the Key Reader is just a mapping technique that takes a sequence of letters that follow a slash and decides from that what—and tells the program from that what menu equivalent it should then execute for that particular action.

Q. I understand.

So when a user selects the Key Reader and makes that active, the Q.EXE program will recognize in running a macro that a slash key is there and will say: Go to that portion of QUATTRO.MU that Mr. Anderson created?

A. It will refer to that portion, yes.

Q. And then the program will work through that portion to find the mapping to the appropriate menu equivalent commands, correct?

A. Yes.

Q. And then it will go back out to Q.EXE to perform whatever is necessary to actually execute that, correct?

A. Correct.

* * *

AFTERNOON SESSION

[143] MR. KONRAD: Q. Mr. Warfield, just before starting here, you came over and checked something in your Quattro Pro for WINDOWS.

Is there an answer to a previous question that you'd like to amend?

A. Quattro Pro for WINDOWS does include the Quattro Pro DOS native menus as an alternate menu you can get when you hit the slash key. So, in fact, you could execute keystroke macros written for the DOS product.

Q. You just couldn't remember that before lunch?

A. Correct.

Q. But looking at the product, you see it there, and it's refreshed your recollection?

A. Yes.

Q. That's fine.

I'd just like to go back briefly to the particular section of QUATTRO.MU that Mr. Anderson created that we were referring to before. I want to make sure that I understand exactly what that section **[144]** contains.

If you looked at a print-out of that section, would you see on one particular line of it a letter corresponding to a menu command from the 123.MU file and then next to it the menu equivalent command that that would correspond to?

A. That's correct.

Q. So you'd see, for example, the letter W, then a menu equivalent command on one separate line?

A. Well, of course, in that particular case, there is no menu equivalent for W, but you might see one next to the T for graph type.

Q. So you'd just see a line containing letters for those menu commands that have a menu equivalent command?

A. There are some letters that simply lead to further menus and have no menu equivalent. The case of W presents the worksheet menu. You have to go down deeper in that menu to find menu equivalents.

Q. In that case, you would see—there would be no line that had just W?

A. There would be a line that had just W, but it would contain no menu equivalents.

[145] Q. What else would be on that line?

A. There wouldn't be anything on that line. Below it would be, for instance, G for worksheet global. G is another one that leads to another menu. So there would be, for instance, G for format. You've got three lines—W, G, F—and it's on the F that contains the menu equivalent for worksheet global format.

Q. But there would be one line corresponding to each menu command in the 123.MU file?

A. I think corresponding to most of them. There may be some cases of commands not present in 1-2-3 that are also not present in the Key Reader.

Q. But for the commands present in 1-2-3, there would be a line corresponding to each?

A. To each menu equivalent, yes. That's correct.

Q. Or corresponding to each menu command?

A. Yes.

Q. And where that command had a menu equivalent, that would appear on the same line. But in some instances, there is no menu equivalent, so all you'd see is the letter?

[146] A. Correct.

Q. And, again, the order of commands, you've told me, is indicated by the outline format.

Are the subsidiary items actually indented if you were to look at a print-out?

A. Yes, they are.

MR. DETKIN: Again, we're referring to the Key Reader portion of the Quattro Pro 123.MU file?

MR. KONRAD: Right, the particular section that we talked about that Mr. Anderson did.

WITNESS: That's my assumption.

MR. KONRAD: Q. And, again, this outline format corresponds to the organization of the menu commands in the 123.MU file?

A. That's correct.

Q. Is there anything else on any one of these individual lines?

A. Not that I recall.

Q. Are these individual lines of code?

A. No, no. They're just—they're just words. It's a mapping.

Q. Is this saved as a text file?

[147] A. It's edited as a text file, but we compile it into an internal representation that's a little more compact.

Q. It is stored within QUATTRO.MU in the compiled form?

A. Yeah. Like I said, the menu equivalents are replaced by numbers, for example, rather than their text.

Q. Are the letters corresponding to menu commands converted into numbers when it's compiled?

A. No, I don't think so, although to be technical about it, the computer can choose to regard them as numbers or as letters.

Q. But to your understanding, they are not in distinction to the menu equivalent commands actually replaced by numbers?

A. That's my understanding.

Q. Now were you responsible for the development of Quattro Pro for WINDOWS?

A. Yes.

Q. And for the Key Reader feature in it?

A. I worked on the design. I was not solely responsible for the Key Reader feature.

[148] Q. Was there any redesign of the Key Reader feature for Quattro Pro for WINDOWS?

A. The concept was the same and the implementation was the same, but we didn't use the same code.

Q. Did the section that Mr . Anderson created survive into Quattro Pro for WINDOWS?

A. I believe it was further edited in order to Quattro Pro for WINDOWS.

Q. Who did that further edit?

A. Barry Spencer.

Q. Do you have any understanding of in what respect it was further edited?

A. I don't know what the exact format of that file is, but I know it is somewhat different from the file used by the DOS program.

Q. When you're referring to "format" there, what format are you talking about?

A. You know, exactly what type of punctuation would need to be used. For instance, every time there's an indentation in the DOS program, the indentation is actually not what signals the next level down to the computer. It's a parentheses that **[149]** does so.

Q. You're saying there's a character in the DOS version of parentheses character which is what actually signals that you're going to another level of the menu?

A. Correct.

Q. The indentation is for ease of viewing?

A. Correct.

Q. And you think perhaps the punctuation sign changed in the WINDOWS version?

A. It may have. I'm just stating I don't know exactly what the format of that file is.

Q. Is it your understanding that Mr. Spencer, whatever he may have done, worked for Mr. Anderson's section?

A. Yes.

Q. Is it your understanding that it is generally—that it contains generally the same information that the DOS version did?

A. Yes, it is.

Q. And, again, is it your understanding that whatever editing Mr. Spencer may have done was performed with it in a text format?

[150] A. Yes. I have seen the file at one point. I just don't recall the exact details of its composition.

Q. Well, of what you saw, did it still have letters on individual lines?

A. Yes.

Q. Was it still indented?

A. I begin to get hazy from there. I recall it was somewhat of an outline format. I saw it briefly when Barry demonstrated for me what he had done. I couldn't tell you any more detail than that.

Q. Do you recall any particular reason that Mr. Spencer was showing you that file?

A. Yeah. He just wanted me to be aware of what he was doing since he had discussed with me what was to be done for the Key Reader.

Q. Do you recall anything else that he told you in terms of any change to that section?

A. No. As I recall, it went something like, "Okay. I've managed to produce what we discussed," and then he showed me a demonstration similar to the one I've described that I gave.

And I said, "Okay. Good job."

* * *

Exhibits in Support of Lotus' Motion for Summary Judgment on Liability, filed May 7, 1991 (Declaration of Kerry L. Konrad (Dkt. Nos. 36 and 148)).

CERTIFICATE OF COPYRIGHT REGISTRATION

FORM TX

UNITED STATES COPYRIGHT OFFICE

OFFICIAL SEAL

This certificate, issued under the seal of the Copyright Office in accordance with the provisions of section 410(a) of title 17, United States Code, attests that copyright registration has been made for the work identified below. The information in this certificate has been made a part of the Copyright Office records.

REGISTER OF COPYRIGHTS
United States of America

REGISTRATION NUMBER

TX 1-233-501
TXU

EFFECTIVE DATE OF REGISTRATION

Dec 23 1983
(Month) (Day) (Year)

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE CONTINUATION SHEET (FORM TX/CON)

1 Title

TITLE OF THIS WORK:

1-2-3

If a periodical or serial give: Vol. ..... No. ..... Issue Date .....

PREVIOUS OR ALTERNATIVE TITLES:

PUBLICATION AS A CONTRIBUTION: (If this work was published as a contribution to a periodical, serial, or collection, give information about the collective work in which the contribution appeared.)

Title of Collective Work: ..... Vol. ..... No. ..... Date ..... Pages .....

2 Author(s)

IMPORTANT: Under the law, the "author" of a "work made for hire" is generally the employer, not the employee (see instructions). If any part of this work was "made for hire" check "Yes" in the space provided, give the employer (or other person for whom the work was prepared) as "Author" of that part, and leave the space for dates blank.

1

NAME OF AUTHOR:

Lotus Development Corporation

Was this author's contribution to the work a "work made for hire"? Yes X No .....

DATES OF BIRTH AND DEATH:

Born ..... (Year) Died ..... (Year)

AUTHOR'S NATIONALITY OR DOMICILE:

Citizen of U.S.A. (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:

Anonymous? Yes ..... No X
Pseudonymous? Yes ..... No X

If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution) entire work (software, documentation, text, illustrations)

2

NAME OF AUTHOR:

Was this author's contribution to the work a "work made for hire"? Yes ..... No .....

DATES OF BIRTH AND DEATH:

Born ..... (Year) Died ..... (Year)

AUTHOR'S NATIONALITY OR DOMICILE:

Citizen of ..... (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:

Anonymous? Yes ..... No .....
Pseudonymous? Yes ..... No .....

If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution)

3

NAME OF AUTHOR:

Was this author's contribution to the work a "work made for hire"? Yes ..... No .....

DATES OF BIRTH AND DEATH:

Born ..... (Year) Died ..... (Year)

AUTHOR'S NATIONALITY OR DOMICILE:

Citizen of ..... (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:

Anonymous? Yes ..... No .....
Pseudonymous? Yes ..... No .....

If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution)

3 Creation and Publication

YEAR IN WHICH CREATION OF THIS WORK WAS COMPLETED:

Year 1982

(This information must be given in all cases.)

DATE AND NATION OF FIRST PUBLICATION:

Date January 26, 1983
(Month) (Day) (Year)

Nation U.S.A.
(Name of Country)

(Complete this block ONLY if this work has been published.)

4 Claimant(s)

NAME(S) AND ADDRESS(ES) OF COPYRIGHT CLAIMANT(S):

Lotus Development Corporation
161 First Street
Boston, Massachusetts

TRANSFER: (If the copyright claimant(s) named here in space 4 are different from the author(s) named in space 2, give a brief statement of how the claimant(s) obtained ownership of the copyright.)

- *Complete all applicable spaces (numbers 5-11) on the reverse side of this page.*
- *Follow detailed instructions attached.* • *Sign the form at line 10.*

J.A. 801

* See TX 1-288-788

TX 1-233-501

(LS Special relief granted under 202.20 (d) of the C. O. reg. "for computer program

** The single page work which accompanies the package contains 1982 and 1983 year-dates in the © notice. the 1982 date was given in error. See memo in file.

EXAMINED BY: HR	APPLICATION RECEIVED: 23 DEC 1983	FOR COPYRIGHT OFFICE USE ONLY
CHECKED BY:		
CORRESPONDENCE: ☑ Yes	DEPOSIT RECEIVED: 23 Dec 1983 * 23 DEC 1983	
DEPOSIT ACCOUNT FUNDS USED: ☐	REMITTANCE NUMBER AND DATE: 74852 UB Special	

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED ADDITIONAL SPACE, USE CONTINUATION SHEET (FORM TX/CON)

PREVIOUS REGISTRATION: (5) Previous Registration

- Has registration for this work, or for an earlier version of this work, already been made in the Copyright Office? Yes ..... No ... X ..
- If your answer is "Yes," why is another registration being sought? (Check appropriate box)
  - ☐ This is the first published edition of a work previously registered in unpublished form.
  - ☐ This is the first application submitted by this author as copyright claimant.
  - ☐ This is a changed version of the work, as shown by line 6 of this application.
- If your answer is "Yes," give Previous Registration Number ..... Year of Registration .....

COMPILATION OR DERIVATIVE WORK: (See instructions) (6) Compilation or Derivative Work

PREEXISTING MATERIAL: (Material, any preexisting work or works that this work is based on or incorporates.)

** .....

MATERIAL ADDED TO THIS WORK: (Give a brief, general statement of the material that has been added to this work and in which copyright is claimed.)

** .....

MANUFACTURERS AND LOCATIONS: (If this is a published work consisting preponderantly of nondramatic literary material in English, the law may require that the copies be manufactured in the United States or Canada for full protection. If so, the names of the manufacturers who performed certain processes, and the places where these processes were performed *must* be given. See instructions for details.) (7) Manufacturing Info.

NAMES OF MANUFACTURERS	PLACES OF MANUFACTURE
Lotus Development Corporation	Cambridge, MA, USA
CSA Inc.	Bedford, MA, USA
Zemag Corp.	Needham, MA, USA
Vinylweld Corp.	Chicago, IL, USA
Apex Paper Box Corp.	Chicago, IL, USA

REPRODUCTION FOR USE OF BLIND OR PHYSICALLY-HANDICAPPED PERSONS: (See instructions) (8) Limitation For Handicapped

- Signature of this form at space 10, and a check in one of the boxes here in space 8, constitutes a non-exclusive grant of permission to the Library of Congress to reproduce and distribute solely for the blind and physically handicapped and under the conditions and limitations prescribed by the regulations of the Copyright Office: (1) copies of the work identified in space 1 of this application in Braille (or similar tactile symbols); or (2) phonorecords embodying a fixation of a reading of that work; or (3) both.

a ☐ Copies and phonorecords b ☐ Copies Only c ☐ Phonorecords Only

DEPOSIT ACCOUNT: (If the registration fee is to be charged to a Deposit Account established in the Copyright Office, give name and number of Account.) (9) Fee made Correspondence address

Name: .....

Account Number: .....

CORRESPONDENCE: (Give name and address to which correspondence about this application should be sent.)

Name: Jerry Cohen, Esq./Cohen & Burg, P.C.

Address: 33 Broad Street

Boston, MA 02109 (Apt.)

(City) 617-742-7840 (State) (ZIP)

CERTIFICATION: * I, the undersigned, hereby certify that I am the: (Check one) (10) Certification

☐ author ☐ other copyright claimant ☐ owner of exclusive right(s) ☒ authorized agent of: Lotus Development Corporation

(Name of author or other copyright claimant, or owner of exclusive right(s))

of the work identified in this application and that the statements made by me in this application are correct to the best of my knowledge.

Handwritten signature: (X) M. A. Fischer

Typed or printed name: Mark A. Fischer Date 12/19/83

MAIL CERTIFICATE TO

Jerry Cohen, Esquire/Cohen & Burg, P.C.

(Name)

33 Broad Street

(Number and Street and Apartment Number)

Boston, Massachusetts 02109

(City) (State) (ZIP code)

17 U.S.C. § 506(e): Any person who knowingly makes a false representation of a material fact in the application for copyright registration provided for by section 409, or in any written statement filed in connection with the application, shall be fined not more than $2,500.

BEST AVAILABLE COPY

CERTIFICATE OF COPYRIGHT REGISTRATION

UNITED STATES COPYRIGHT OFFICE • THE LIBRARY OF CONGRESS •

OFFICIAL SEAL

This certificate, issued under the seal of the Copyright Office in accordance with the provisions of section 410(a) of title 17, United States Code, attests that copyright registration has been made for the work identified below. The information in this certificate has been made a part of the Copyright Office records.

REGISTER OF COPYRIGHTS
United States of America

FORM TX
UNITED STATES COPYRIGHT OFFICE

REGISTRATION NUMBER
TX 1-233-502
(TX) TXU

EFFECTIVE DATE OF REGISTRATION
Dec 23 1983
(Month) (Day) (Year)

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE CONTINUATION SHEET (FORM TX/CON)

1 Title

TITLE OF THIS WORK:
1-2-3 Release 1A

PREVIOUS OR ALTERNATIVE TITLES:

If a periodical or serial give: Vol. ..... No. ..... Issue Date .....

PUBLICATION AS A CONTRIBUTION: (If this work was published as a contribution to a periodical, serial, or collection, give information about the collective work in which the contribution appeared.)

Title of Collective Work: ..... Vol. ..... No. ..... Date ..... Pages .....

2 Author(s)

IMPORTANT: Under the law, the "author" of a "work made for hire" is generally the employer, not the employee (see instructions). If any part of this work was "made for hire" check "Yes" in the space provided, give the employer (or other person for whom the work was prepared) as "Author" of that part, and leave the space for dates blank.

1

NAME OF AUTHOR:
Lotus Development Corporation
Was this author's contribution to the work a "work made for hire"? Yes X ... No .....

DATES OF BIRTH AND DEATH:
Born ..... (Year) Died ..... (Year)

AUTHOR'S NATIONALITY OR DOMICILE:
Citizen of U.S.A. (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:
Anonymous? Yes ..... No X ...
Pseudonymous? Yes ..... No X ...
If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution)
entire work (software, documentation, text, illustrations)

2

NAME OF AUTHOR:

DATES OF BIRTH AND DEATH:
Born ..... (Year) Died ..... (Year)

Was this author's contribution to the work a "work made for hire"? Yes ..... No .....

AUTHOR'S NATIONALITY OR DOMICILE:
Citizen of ..... (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:
Anonymous? Yes ..... No .....
Pseudonymous? Yes ..... No .....
If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution)

3

NAME OF AUTHOR:

DATES OF BIRTH AND DEATH:
Born ..... (Year) Died ..... (Year)

Was this author's contribution to the work a "work made for hire"? Yes ..... No .....

AUTHOR'S NATIONALITY OR DOMICILE:
Citizen of ..... (Name of Country) } or { Domiciled in ..... (Name of Country)

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK:
Anonymous? Yes ..... No .....
Pseudonymous? Yes ..... No .....
If the answer to either of these questions is "Yes," see detailed instructions attached.

AUTHOR OF: (Briefly describe nature of this author's contribution)

3 Creation and Publication

YEAR IN WHICH CREATION OF THIS WORK WAS COMPLETED:
Year 1983
(This information must be given in all cases.)

DATE AND NATION OF FIRST PUBLICATION:
Date June 17, 1983
(Month) (Day) (Year)
Nation U.S.A.
(Name of Country)
(Complete this block ONLY if this work has been published.)

4 Claimant(s)

NAME(S) AND ADDRESS(ES) OF COPYRIGHT CLAIMANT(S):
Lotus Development Corporation
161 First Street
Cambridge, Massachusetts 02142

TRANSFER: (If the copyright claimant(s) named here in space 4 are different from the author(s) named in space 2, give a brief statement of how the claimant(s) obtained ownership of the copyright.)

• Complete all applicable spaces (numbers 5-11) on the reverse side of this page
• Follow detailed instructions attached • Sign the form at line 10

DO NOT WRITE HERE

J.A. 803

**Special relief granted under 202.20 (d) of the C. O. regs for the computer program

TX, 1-233-502

EXAMINED BY: NR
CHECKED BY:

APPLICATION RECEIVED: 23 DEC 1983

CORRESPONDENCE: ☒ Yes

DEPOSIT RECEIVED: 23 Dec 1983 23 DEC 1983

DEPOSIT ACCOUNT FUNDS USED: ☐

REMITTANCE NUMBER AND DATE: 74852 UB Special

FOR COPYRIGHT OFFICE USE ONLY

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED ADDITIONAL SPACE, USE CONTINUATION SHEET (FORM TX/CON)

PREVIOUS REGISTRATION: (5) Previous Registration

- Has registration for this work, or for an earlier version of this work, already been made in the Copyright Office? Yes ..... No ..X....
- If your answer is "Yes," why is another registration being sought? (Check appropriate box)
  - ☐ This is the first published edition of a work previously registered in unpublished form.
  - ☐ This is the first application submitted by this author as copyright claimant.
  - ☐ This is a changed version of the work, as shown by line 6 of this application.
- If your answer is "Yes," give: Previous Registration Number ..... Year of Registration .....

COMPILATION OR DERIVATIVE WORK: (See instructions) (6) Compilation or Derivative Work

PREEXISTING MATERIAL. Identify any preexisting work or works that this work is based on or incorporates.
1-2-3 (earlier version)

MATERIAL ADDED TO THIS WORK. Give a brief, general statement of the material that has been added to this work and in which copyright is claimed.
editorial revisions in program

MANUFACTURERS AND LOCATIONS: (If this is a published work consisting preponderantly of nondramatic literary material in English, the law may require that the copies be manufactured in the United States or Canada for full protection. If so, the names of the manufacturers who performed certain processes, and the places where these processes were performed must be given. See instructions for details.) (7) Manufacturing

NAMES OF MANUFACTURERS	PLACES OF MANUFACTURE
Lotus Development Corporation	Cambridge, MA, USA
CSA, Inc.	Bedford, MA, USA
Zemag Corp.	Needham, MA, USA
Vinylweld Corp.	Chicago, IL, USA
Apex Paper Box Corp.	Chicago, IL, USA

REPRODUCTION FOR USE OF BLIND OR PHYSICALLY-HANDICAPPED PERSONS: (See instructions) (8) License - For Handicapped

- Signature of this form at space 10, and a check in one of the boxes here in space 8, constitutes a non-exclusive grant of permission to the Library of Congress to reproduce and distribute solely for the blind and physically handicapped and under the conditions and limitations prescribed by the regulations of the Copyright Office: (1) copies of the work identified in space 1 of the application in Braille (or similar tactile symbols); or (2) phonorecords embodying a fixation of a reading of that work; or (3) both.

a ☐ Copies and phonorecords b ☐ Copies Only c ☐ Phonorecords Only

DEPOSIT ACCOUNT: (If the registration fee is to be charged to a Deposit Account established in the Copyright Office, give name and number of Account.)

Name: .....
Account Number: .....

CORRESPONDENCE: (Give name and address to which correspondence about this application should be sent.) (9) Fee and Correspondence

Name: Jerry Cohen, Esq./Cohen & Burg, P.C.
Address: 33 Broad Street
Boston, MA 02109
(City) (State) (ZIP)
(617) 742-7840 (Tel.)

CERTIFICATION: * I, the undersigned, hereby certify that I am the: (Check one) (10) Certification (Application must be signed)

☐ author ☐ other copyright claimant ☐ owner of exclusive right(s) ☒ authorized agent of: Lotus Development Corporation
(Name of author or other copyright claimant, or owner of exclusive right(s))

of the work identified in this application and that the statements made by me in this application are correct to the best of my knowledge.

Handwritten signature: Mark A. Fischer

Typed or printed name: Mark A. Fischer Date: 12/19/83

(11) Address For Return of Certificate

Jerry Cohen, Esquire/Cohen & Burg, P.C.
(Name)
33 Broad Street
(Number, Street and Apartment Number)
Boston, Massachusetts 02109
(City) (State) (ZIP code)

MAIL CERTIFICATE TO

(Certificate will be mailed in window envelope)

CERTIFICATE OF COPYRIGHT REGISTRATION

This certificate, issued under the seal of the Copyright Office in accordance with the provisions of section 410(a) of title 17, United States Code, attests that copyright registration has been made for the work identified below. The information in this certificate has been made a part of the Copyright Office records.

OFFICIAL SEAL

REGISTER OF COPYRIGHTS
United States of America

FORM TX
UNITED STATES COPYRIGHT OFFICE

REGISTRATION NUMBER

TX 1 694 602

TX TXU

EFFECTIVE DATE OF REGISTRATION

11 (Month) 8 (Day) 85 (Year)

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE A SEPARATE CONTINUATION SHEET.

1 TITLE OF THIS WORK ▼

1-2-3 RELEASE 2

PREVIOUS OR ALTERNATIVE TITLES ▼

PUBLICATION AS A CONTRIBUTION If this work was published as a contribution to a periodical, serial, or collection, give information about the collective work in which the contribution appeared. Title of Collective Work ▼

If published in a periodical or serial give: Volume ▼ Number ▼ Issue Date ▼ On Pages ▼

2 NAME OF AUTHOR ▼

a LOTUS DEVELOPMENT CORPORATION

DATES OF BIRTH AND DEATH
Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☒ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of Country
OR { Citizen of ▶ ____ Domiciled in ▶ United States

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK
Anonymous? ☐ Yes ☐ No
Pseudonymous? ☐ Yes ☐ No
If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼ Manual, Tutorial, Getting Started, Quick Reference and computer program text

NOTE
Under the law, the "author" of a

NAME OF AUTHOR ▼

DATES OF BIRTH AND DEATH
Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of Country
OR { Citizen of ▶ ____ Domiciled in ▶ ____

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK
Anonymous? ☐ Yes ☐ No
Pseudonymous? ☐ Yes ☐ No
If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

NAME OF AUTHOR ▼

DATES OF BIRTH AND DEATH
Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of Country
OR { Citizen of ▶ ____ Domiciled in ▶ ____

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK
Anonymous? ☐ Yes ☐ No
Pseudonymous? ☐ Yes ☐ No
If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

3 YEAR IN WHICH CREATION OF THIS WORK WAS COMPLETED This information must be given in all cases. 1985 ◀ Year

DATE AND NATION OF FIRST PUBLICATION OF THIS PARTICULAR WORK
Complete this information ONLY if this work has been published. Month ▶ September Day ▶ 25 Year ▶ 1985
United States ◀ Nation

4 COPYRIGHT CLAIMANT(S) Name and address must be given even if the claimant is the same as the author given in space 2. ▼

LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142

See instructions before completing this space.

TRANSFER If the claimant(s) named here in space 4 are different from the author(s) named in space 2, give a brief statement of how the claimant(s) obtained ownership of the copyright. ▼

DO NOT WRITE HERE OFFICE USE ONLY

APPLICATION RECEIVED
NOV 08 1985

ONE DEPOSIT RECEIVED

TWO DEPOSITS RECEIVED
NOV 08 1985

REMITTANCE NUMBER AND DATE

00145665 4

J.A. 805

EXAMINED BY

CHECKED BY

FORM TX

TX 1 694 602

☒ CORRESPONDENCE Yes

☐ DEPOSIT ACCOUNT FUNDS USED

FOR COPYRIGHT OFFICE USE ONLY

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE A SEPARATE CONTINUATION SHEET.

PREVIOUS REGISTRATION Has registration for this work, or for an earlier version of this work, already been made in the Copyright Office? **5**

☒ Yes ☐ No If your answer is "Yes," why is another registration being sought? (Check appropriate box) ▼

☐ This is the first published edition of a work previously registered in unpublished form.

☐ This is the first application submitted by this author as copyright claimant.

☒ This is a changed version of the work, as shown by space 6 on this application.

If your answer is "Yes," give: Previous Registration Number ▼ Year of Registration ▼

TX 1-233-501 TX 1-233-502 1983

DERIVATIVE WORK OR COMPILATION Complete both space 6a & 6b for a derivative work; complete only 6b for a compilation. **6**

a. Preexisting Material Identify any preexisting work or works that this work is based on or incorporates. ▼

1-2-3 1-2-3 Release 1A

Owner of copyrights in preexisting works

b. Material Added to This Work Give a brief, general statement of the material that has been added to this work and in which copyright is claimed. ▼

Program revised and updated; Instructional materials modified accordingly

See instructions before completing this space

MANUFACTURERS AND LOCATIONS If this is a published work consisting preponderantly of nondramatic literary material in English, the law may require that the copies be manufactured in the United States or Canada for full protection. If so, the names of the manufacturers who performed certain processes, and the places where these processes were performed must be given. See instructions for details. **7**

Names of Manufacturers ▼ Eusey Press — Places of Manufacture ▼ United States

PRODUCTION FOR USE OF BLIND OR PHYSICALLY HANDICAPPED INDIVIDUALS A signature on this form at space 10, and a check in one of the boxes here in space 8, constitutes a non-exclusive grant of permission to the Library of Congress to reproduce and distribute solely for the blind and physically handicapped and under the conditions and limitations prescribed by the regulations of the Copyright Office: (1) copies of the work identified in space 1 of this application in Braille (or similar tactile symbols); or (2) phonorecords embodying a fixation of a reading of that work; or (3) both. **8**

a ☐ Copies and Phonorecords b ☐ Copies Only c ☐ Phonorecords Only

See instructions

DEPOSIT ACCOUNT If the registration fee is to be charged to a Deposit Account established in the Copyright Office, give name and number of Account. **9**

Name ▼ SPENCER & FRANK Account Number ▼ DA 0 33669

CORRESPONDENCE Give name and address to which correspondence about this application should be sent. Name/Address/Apt/City/State/Zip ▼

Mark Harrison c/o Spencer & Frank
1111 19th Street N.W.
Washington, D.C. 20036

Area Code & Telephone Number ▶ 202 828 8000

Be sure to give your daytime phone number

CERTIFICATION* I, the undersigned, hereby certify that I am the **10**

Check one ▶
☐ author
☐ other copyright claimant
☐ owner of exclusive right(s)
☒ authorized agent of LOTUS DEVELOPMENT CORP.
Name of author or other copyright claimant, or owner of exclusive right(s) ▲

of the work identified in this application and that the statements made by me in this application are correct to the best of my knowledge.

Typed or printed name and date ▼ If this is a published work, this date must be the same as or later than the date of publication given in space 3.

MARK HARRISON date ▶ 11/07/85

Handwritten signature (X) ▼

Mark Harrison

MAIL CERTIFICATE TO Certificate will be mailed in window envelope

Name ▼ Mark Harrison c/o Spencer & Frank

Number/Street/Apartment Number ▼ 1111 19th Street N.W.

City/State/ZIP ▼ Washington, D.C. 20036

11

Have you:
- Completed all necessary spaces?
- Signed your application in space 10?
- Enclosed check or money order for $10 payable to Register of Copyrights?
- Enclosed your deposit material with the application and fee?

MAIL TO: Register of Copyrights, Library of Congress, Washington, D.C. 20559

* 17 U.S.C. § 506(e): Any person who knowingly makes a false representation of a material fact in the application for copyright registration provided for by section 409, or in any written statement filed in connection with the application, shall be fined not more than $2,500.

☆ GPO 1984—421-278/513

May 1984—75,000

UNITED STATES COPYRIGHT OFFICE
LIBRARY OF CONGRESS

OFFICIAL SEAL

This certificate, issued under the seal of the Copyright Office in accordance with the provisions of section 410(a) of title 17, United States Code, attests that copyright registration has been made for the work identified below. The information in this certificate has been made a part of the Copyright Office records.

REGISTER OF COPYRIGHTS
United States of America

REGISTRATION NUMBER

TX 2 005 501

TX TXU

EFFECTIVE DATE OF REGISTRATION

MARCH 23 1987
Month Day Year

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE A SEPARATE CONTINUATION SHEET.

1 **TITLE OF THIS WORK ▼**

1-2-3 Release 2.01 5.25" Disk Version

PREVIOUS OR ALTERNATIVE TITLES ▼

PUBLICATION AS A CONTRIBUTION If this work was published as a contribution to a periodical, serial, or collection, give information about the collective work in which the contribution appeared. **Title of Collective Work ▼**

If published in a periodical or serial give: Volume ▼ **Number ▼** **Issue Date ▼** **On Pages ▼**

2 **a** **NAME OF AUTHOR ▼**

Lotus Development Corporation

DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☒ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of Country OR { Citizen of ▶ ____ Domiciled in ▶ U.S.A.

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☒ No Pseudonymous? ☐ Yes ☒ No If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

text and illustrations of Getting Started, Tutorial, Reference Manual, Quick Reference, computer program text, audiovisual display and template text

NOTE Under the law, the "author" of a "work made for hire" is generally the employer, not the employee (see instructions). For any part of this work that was "made for hire" check "Yes" in the space provided, give the employer (or other person for whom the work was prepared) as "Author" of that part, and leave the space for dates of birth and death blank.

b **NAME OF AUTHOR ▼**

DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of country OR { Citizen of ▶ ____ Domiciled in ▶ ____

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☐ No Pseudonymous? ☐ Yes ☐ No If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

c **NAME OF AUTHOR ▼**

DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼

Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No

AUTHOR'S NATIONALITY OR DOMICILE Name of Country OR { Citizen of ▶ ____ Domiciled in ▶ ____

WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☐ No Pseudonymous? ☐ Yes ☐ No If the answer to either of these questions is "Yes," see detailed instructions.

NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

3 **YEAR IN WHICH CREATION OF THIS WORK WAS COMPLETED** This information must be given in all cases. 1986 ◀ Year

DATE AND NATION OF FIRST PUBLICATION OF THIS PARTICULAR WORK Complete this information ONLY if this work has been published. Month ▶ 11 Day ▶ 17 Year ▶ 1986 U.S.A. ◀ Nation

4 **COPYRIGHT CLAIMANT(S)** Name and address must be given even if the claimant is the same as the author given in space 2.▼

Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142

See instructions before completing this space

TRANSFER If the claimant(s) named here in space 4 are different from the author(s) named in space 2, give a brief statement of how the claimant(s) obtained ownership of the copyright.▼

DO NOT WRITE HERE OFFICE USE ONLY

APPLICATION RECEIVED
MAR 23 1987

ONE DEPOSIT RECEIVED
MAR 23 1987

TWO DEPOSITS RECEIVED

REMITTANCE NUMBER AND DATE

MORE ON BACK ▶ • Complete all applicable spaces (numbers 5-11) on the reverse side of this page. • See detailed instructions. • Sign the form at line 10.

DO NOT WRITE HERE

J.A. 807

TA 2 005 501

☒ CORRESPONDENCE Yes

☐ DEPOSIT ACCOUNT FUNDS USED

FOR COPYRIGHT OFFICE USE ONLY

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE A SEPARATE CONTINUATION SHEET.

PREVIOUS REGISTRATION Has registration for this work, or for an earlier version of this work, already been made in the Copyright Office?

☒ Yes ☐ No If your answer is "Yes," why is another registration being sought? (Check appropriate box) ▼

5

☐ This is the first published edition of a work previously registered in unpublished form.

☐ This is the first application submitted by this author as copyright claimant.

☒ This is a changed version of the work, as shown by space 6 on this application.

If your answer is "Yes," give: Previous Registration Number ▼ Year of Registration ▼

TX 1-694-602 1985 / TX 1-233-501, TX 1-233-502 1983

DERIVATIVE WORK OR COMPILATION Complete both space 6a & 6b for a derivative work; complete only 6b for a compilation.

a. Preexisting Material Identify any preexisting work or works that this work is based on or incorporates. ▼

1-2-3; 1-2-3 Release 1A; 1-2-3 Release 2

6

b. Material Added to This Work Give a brief, general statement of the material that has been added to this work and in which copyright is claimed. ▼

Program revised and updated; instructional materials modified accordingly.

See instructions before completing this space.

MANUFACTURERS AND LOCATIONS If this is a published work consisting preponderantly of nondramatic literary material in English, the law may require that the copies be manufactured in the United States or Canada for full protection. If so, the names of the manufacturers who performed certain processes, and the places where these processes were performed must be given. See instructions for details.

7

Names of Manufacturers ▼	Places of Manufacture ▼
The Eusey Press	Leominster, Massachusetts, U.S.A.
The Alpine Press	Stoughton, Massachusetts, U.S.A.

REPRODUCTION FOR USE OF BLIND OR PHYSICALLY HANDICAPPED INDIVIDUALS A signature on this form at space 10, and a check in one of the boxes here in space 8, constitutes a non-exclusive grant of permission to the Library of Congress to reproduce and distribute solely for the blind and physically handicapped and under the conditions and limitations prescribed by the regulations of the Copyright Office: (1) copies of the work identified in space 1 of this application in Braille (or similar tactile symbols); or (2) phonorecords embodying a fixation of a reading of that work; or (3) both.

8

a ☐ Copies and Phonorecords b ☐ Copies Only c ☐ Phonorecords Only

See instructions.

DEPOSIT ACCOUNT If the registration fee is to be charged to a Deposit Account established in the Copyright Office, give name and number of Account.

9

Name ▼	Account Number ▼
Lotus Development Corporation	DA 044210

CORRESPONDENCE Give name and address to which correspondence about this application should be sent. Name/Address/Apt/City/State/Zip ▼

Janis M. LaRue, Esq., c/o Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142

Area Code & Telephone Number ▶ (617) 577-8500

Be sure to give your daytime phone number.

CERTIFICATION* I, the undersigned, hereby certify that I am the

Check one ▶
☐ author
☐ other copyright claimant
☐ owner of exclusive right(s)
☒ authorized agent of Lotus Development Corporation

Name of author or other copyright claimant, or owner of exclusive right(s) ▲

10

of the work identified in this application and that the statements made by me in this application are correct to the best of my knowledge.

Typed or printed name and date ▼ If this is a published work, this date must be the same as or later than the date of publication given in space 3.

Janis M. LaRue date ▶ 3/20/87

Handwritten signature (X) ▼ J. M. LaRue

MAIL CERTIFICATE TO

Certificate will be mailed in window envelope

Name ▼
Janis M. LaRue, Esq.
Lotus Development Corporation

Number/Street/Apartment Number ▼
55 Cambridge Parkway

City/State/ZIP ▼
Cambridge, MA 02142

11

Have you:
- Completed all necessary spaces?
- Signed your application in space 10?
- Enclosed check or money order for $10 payable to *Register of Copyrights*?
- Enclosed your deposit material with the application and fee?

MAIL TO: Register of Copyrights, Library of Congress, Washington, D.C. 20559.

* 17 U.S.C. § 506(e): Any person who knowingly makes a false representation of a material fact in the application for copyright registration provided for by section 409, or in any written statement filed in connection with the application, shall be fined not more than $2,500.

• U.S. GOVERNMENT PRINTING OFFICE: 1984: 421-278/10,006

October 1984—50,000

EXHIBIT 26

NEWS

For Immediate Release

Contact
James Strohecker
Cunningham Communication, Inc.
(408) 982-0400

Tami Casey
Borland International, Inc.
(408) 438-8400

BORLAND INTERNATIONAL ANNOUNCES THE GREAT AMERICAN QUATTRO UPGRADE

30-Day Offer Provides 60 Percent Off Spreadsheet Software to Lotus 1-2-3 Users

SCOTTS VALLEY, Calif.—October 13, 1988—Borland International, Inc. today announced a limited-time offer for its popular Quattro spreadsheet to Lotus 1-2-3 users for only $99.95. The program, called the Great American Quattro Upgrade, requires 1-2-3 owners to send in the front cover of their user's manual and $99.95 in exchange for the current version of Quattro. The offer is restricted to the U.S. and Canada and will run from October 7 until November 8, 1988.

"The recent delay of 1-2-3 Version 3 is a great reason for spreadsheet users to buy Quattro, and now we are giving even more incentive to upgrade, with twice the performance at a fraction of the price," said Philippe Kahn, Borland chairman, president and chief executive officer. "This gives end users an opportunity to easily and cost-effectively switch to a company that develops products on time, with demonstrated technological and price/performance advantages. We expect this promotion to stimulate even greater market interest in Quattro and heighten retail demand for the product."

The limited time offer is only available directly through Borland. Mailed requests must be postmarked before Novem-

ber 9, 1988. Customers should call the customer service line at (408) 438-8400 for further information.

Borland International, Inc. is a leading developer of high-performance microcomputer software products that match power with ease of learning and use. The Scotts Valley, Calif.-based company, founded in 1983, ranks sixth among the leading U.S. microcomputer software companies. Borland's stock is publicly traded in the United Kingdom. Borland offers some of the world's most widely used software business applications and developer's tools such as Paradox, Quattro, SideKick Plus and the Turbo Language Series.

###

Paradox, Quattro and SideKick Plus, Sprint and the Turbo Language Series are registered trademarks of Borland International, Inc.

Lotus and 1-2-3 are registered trademarks of Lotus Corporation.

EXHIBIT 30

N E W S

FOR IMMEDIATE RELEASE

Contact:
Catherine Miller
Borland International, Inc.
(408) 439-1614

BORLAND'S QUATTRO PRO WINS OVER LOTUS 1-2-3 USERS

Survey Results Indicate Exceptional Levels of Owner Satisfaction

SCOTTS VALLEY, Calif.—April 17, 1990—Quattro Pro from Borland International, Inc. (NASDAQ:BORL) is becoming the high-end spreadsheet of choice for many experienced spreadsheet users, according to a study conducted by Vista Marketing Research, an independent market research firm located in Oakland, Calif.

"The percentage of respondents rating the product as excellent or very good was exceptional and shows a very-high level of consumer satisfaction," said Thomas E. McCarty, president of Vista Marketing Research.

To determine attitudes about Quattro Pro, as well as upgrade and usage patterns, Vista Marketing Research mailed questionnaires to a randomly selected group of spreadsheet users who have upgraded to Quattro Pro from Lotus 1-2-3, Microsoft Excel, Computer Associates' SuperCalc, Lotus Symphony or the original Quattro. The samples were taken from groups of upgrade customers who had purchased Quattro Pro at least 60 days before the date the questionnaires were sent out. The results are based on 411 completed returns during January-March 1990.

SUMMARY OF SURVEY FINDINGS

* 89% of the respondents rated Quattro Pro "Excellent" or "Very Good"
* 93% are continuing to use Quattro Pro
* 83% said they have discontinued using their previous spreadsheet and are using Quattro Pro as their sole spreadsheet program
* 89% were previously using Symphony or Lotus 1-2-3 (all versions)
* Some respondents upgraded twice—
  18% from 1-2-3 release 2.2
  10% from 1-2-3 release 3.0
* Quattro Pro is being used on high-end machines—
  38% reported using Quattro Pro on 386 PCs
  48% are using Quattro Pro on 286 PCs
* Quattro Pro is being used by experienced spreadsheet users
  36% classified themselves as "power" users
  59% said they were light/medium users
  5% of the respondents classified themselves as novice spreadsheet users

Borland International, Inc. (Scotts Valley, Calif.) is a leading developer of high performance software products, including Paradox, Quattro Pro, SideKick, Turbo Pascal, Turbo C and Turbo Debugger & Tools.

###

QUATTRO PRO® OWNER SATISFACTION
(411 Respondents: Jan-Mar.1990)

- 89% rated Quattro Pro "Excellent/Very Good"
- 93% continue to use Quattro Pro
- 83% claim to have discarded their previous spreadsheet
- 89% switched from a Lotus product
  (Symphony® and all versions of 1-2-3®)
- Several respondents apparently upgraded twice—
  18% switched from 1-2-3 version 2.2
  10% switched from 1-2-3 version 3.0
- Quattro Pro is used on high-end machines—
  38% use Quattro Pro on 386 PCs
  48% use Quattro Pro on 286 PCs
- Quattro Pro users are experienced spreadsheet users—
  59% Light/Medium Users
  36% Power Users
  5% Novice Users

Source: Vista Marketing Research, Oakland, CA
(415)-839-0256
Contact: Thomas E. McCarty, President

EXHIBIT C-27

CONFIDENTIAL

B O R L A N D
1800 Green Hills Road
Scotts Valley, CA 95067

Try the top-rated spreadsheet.
At a rock-bottom price.

Offer good until
July 30, 1990.

Marc Randolph
Borland International
Dept. BI634-CA
1700 Green Hills Road
Scotts Valley, CA 95067

Dear Spreadsheet User:

Borland International would like to make you an unprecedented offer.

- For a limited time, you can upgrade from your current spreadsheet to Borland's extraordinary QUATTRO® PRO for only $99.95—you save $395 off the suggested retail price.
- If you choose, we'll bill you when the program is delivered.
- You'll receive a spreadsheet that has gotten rave reviews in the computer press. *PC Magazine* said, "Seeing QUATTRO PRO in action is believing: this is the top-dog spreadsheet." And *InfoWorld* said, "This spreadsheet may just be the best available in terms of performance, capacity, (and) ease of use."

- If you've been using Lotus® 1-2-3®, you'll have no learning curve. QUATTRO PRO is so compatible with 1-2-3 that you can be up and running in just 10 minutes.
- You also get a *60-day money-back guarantee*. If you're not pleased with QUATTRO PRO for any reason, return it to us for a full refund.

For just $99.95, QUATTRO PRO offers you the most important advanced features of 1-2-3 version 2.2, 1-2-3 version 3.0, Excel® and SuperCalc®—all in one program, *without requiring extensive new PC hardware*. You'll enjoy:

- *Exceptional 1-2-3 compatibility.* You can direct QUATTRO PRO to accept the same keystroke procedures, menu commands and macros as 1-2-3. And QUATTRO PRO lets you read, write and load even the largest files created with 1-2-3 Releases 1A and 2.01—*without translation*.
- *Multi-page spreadsheet consolidation.* Take any cells in the spreadsheet you're using and create "hot links" to cells in up to 64 other spreadsheets, graphs and databases—loaded *and* unloaded. This means you can reference *any* cell from *any* spreadsheet from *any* formula you're working in.

(over, please)

Exhibit to Lotus' Motion for Summary Judgment, filed November 1, 1991 (Declaration of Paul M. O'Connor III (Dkt. No. 108))

EXHIBIT C

UNITED STATES COURT OF APPEALS
FOR THE NINTH CIRCUIT

No. 89-15683

Filed Dec. 6, 1989

ASHTON-TATE CORPORATION, a Delaware corporation,

Appellee/Plaintiff,

vs.

RICHARD ROSS; BRAVO TECHNOLOGIES, INC.,
a California corporation,

Appellants/Defendants.

RICHARD A. ROSS; BRAVO TECHNOLOGIES, INC.,
a California corporation,

Counterclaimants-Appellants,

vs.

ASHTON-TATE CORPORATION,
and DOES ONE through FIVE,

Counterdefendant-Appellee.

J.A. 818

On Appeal from the Order of the United States
District Court for the Northern District of California
No. C-88-2844 DLJ

Honorable D. Lowell Jensen, District Judge

APPELLEE'S BRIEF

GARY L. REBACK
CYNTHIA C. CANNADY
LESLIE A. FITHIAN
FENWICK, DAVIS & WEST
2 Palo Alto Square, Suite 800
Palo Alto, CA 94306
Telephone (415)494-0600

Attorneys for Appellee/Plaintiff
ASHTON-TATE CORPORATION

---

* * *

[18] features on Ross' list. Ross Depo., CR 40, at 142-43. Although Appellant's Brief repeatedly refers to the two exhibits together as constituting "a detailed layout for a major portion of the user interface of Full Impact," *see, e.g.*, App. Opening Brief at 24, it is undisputed that Ross' only "contribution" was Exhibit 8. Ross Depo., CR 40, at 103-05. Wigginton, by Ross' own admission, prepared the list and description of features identified as Exhibit 9. Ross Depo., CR 40, at 142-43.

At his deposition, Ross was asked to use colored markers to highlight those commands on his list (Exhibit 8) that are

standard Apple commands and standard spreadsheet commands. He highlighted the standard Apple commands in pink and the standard spreadsheet commands in yellow. He then admitted that the commands that remained unhighlighted on his list (Exhibit 8) are not even present in Full Impact. Ross Depo., CR 40, at 111-40. A copy of this highlighted list as it was presented to the Trial Court with an accompanying declaration, is found at Supplemental Excerpts, Tab 40, Ross Depo. Ex. 8.

Referring to this highlighted list, Ross testified as follows:

> Q. You're not making any claim to the ones that are marked in pink?
> A. No, not individually.
> Q. You're not making any claims to the ones marked in yellow?
> A. Not item by item, no.
> Q. Are you making any claim that the way these things are arranged can be proprietary to you?
> A. All of the menus, certainly, yes.
> Q. What you think is proprietary is the way you've arranged nonproprietary commands on one piece of paper; is that correct?
> A. Yes.

Ross Depo., CR 40, at 139-40.

But Ross thereafter admitted that the arrangement of the commands in Full Impact is not the same as the order in which they appear on his list, Exhibit 8, that Full Impact does not use all of the commands on his list, and

* * *

[37] attaches if the requisite "originality" is found in the selection and arrangement process, *Financial Information v. Moody's Investors Service*, 231 U.S.P.Q. 803, 805, *aff'd*, 808 F.2d 204 (2d Cir. 1986), *cert. denied*, 484 U.S. 820 (1987), and in any case, the protection does not "imply any exclusive right in the preexisting material," 17 U.S.C. § 103(a), as opposed to the arrangement itself. *Harper House, supra.*

Here Ross' selection and arrangement of noncopyrightable elements was dictated largely by the Macintosh interface guidelines and standard spreadsheet commands. Wigg. Decl., CR 38, ¶ 7; Ross Decl., CR 53, ¶ 10; Ross Depo., CR 40, at 315-16. As was the case in *Financial Information, supra* at 805, Ross' "coordination and arrangement" consisted only of placing the uncopyrightable elements "in a uniform mold" dictated by an "externality." Accordingly, copyright "originality" is absent. *Id.*; *cf. Whelan Associates v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1237 (3d Cir. 1987).

In addition, the arrangement and selection of these commands in the Full Impact product is different from that of Ross' list. Full Impact contains commands absent from Ross' list. Ross' list contains commands absent from Full Impact. The only commands common to both are in an arrangement, order and grouping in Full Impact different from that of Ross' list. Ross Depo., CR 40, at 111-40, 315-16, 321; Wigg. Decl., CR 38, ¶ 15 & Ex. B. Even if Exhibit 8 constituted a copyrightable "compilation," which by Ross' own testimony it does not, Ross would have no interest in the pre-existing elements of the compilation as opposed to the arrangement itself. 17 U.S.C. § 103(a). Since the arrangement is admittedly different in Full Impact, Ross has no claim for relief. *See Harper House, Inc. v. Thomas Nelson, Inc.*, Nos. 87-6398, 876411. 1988 U.S. App. Lexis 16284 (9th Cir. 1989).

Finally, even if Exhibit 8 were copyrightable expression, Ross' contribution of Exhibit 8 to the MacCalc prototype would not make him a joint

* * *

Exhibits in Support of Borland's Cross-Motion for Summary Judgment on the Issue of Copyrightability and in Opposition to Lotus' Motion for Summary Judgment, dated January 30, 1992 (Dkt. No. 142)

EXHIBIT 1

QUATTRO PRO HONORS: AWARDS AND QUOTES

Quattro Pro's state-of-the-art technology, innovative features, superior graphics and modern user interface have made it incredibly popular. In just over one year, Quattro Pro has earned over 30 product awards from reviewers and users worldwide. A listing of this unprecedented collection of awards is followed by an assortment of quotes pulled from various PC industry and business publications.

PC MAGAZINE
— Award for Technical Excellence, November 1989
— Editor's Choice, December 1990
— Editor's Choice, April 1990

PC MAGAZINE (UK)
— Editor's Choice, December 1990

BYTE MAGAZINE
— Award of Distinction, December 1989
— Reader's VIP Award, Spreadsheet Category, June 1990
— 1990 Award of Excellence, January 1991
— 1991 Reader's Choice Awards, Spreadsheet Category, July 1991

INFOWORLD
— Best in Its Class, January 1990
— Top 100 Products of 1989, February 1990
— Overall MS-DOS Software Product of the Year, March 1990
— Best Spreadsheet Software, March 1990
— InfoWorld Product of the Year 1990, February 1991
— Buyer's Assurance Seal, August 1991

DATAMATION
— 3rd place winner 1990 Software Products of the Year, February 1991
— Finalist 1990 Software Products of the Year, October 1990

Copyright 1991, Borland International. All rights reserved.

PERSONAL COMPUTER MAGAZINE (PCM)—UK
— Editor's Choice, January 1990

PC/COMPUTING
— Most Valuable Product of 1989, January 1990

PC WEEK
— Top Score in a special spreadsheet comparison, March 1990

PC WORLD
— Best Buy, April 1990
— 1990 World Class Award, Promising Newcomer Software, September 1990

PC WORLD—UK
— World Class Award, Spreadsheets, December 1990

PC WORLD—DENMARK
— Top-Rated Spreadsheet, March 1990

PC WORLD—NEW ZEALAND
— Editor's Choice In High End Spreadsheet Comparison, October 1990

PC USER—UK
— 1989 Software Product of the Year, April 1990
— Runner-up, Best New Program, April 1990

SOFTWARE PUBLISHER'S ASSOCIATION (SPA)
— Best Business Application: Numeric or Data Orientation, March 1990
— Best Business Application: Numeric or Data Orientation, March 1991
— Critics Choice Award: Best Business Application Finalist, March 1991

RESELLER MANAGEMENT
— 1990 Readers' Choice Award, October 1990

AUSTRALIAN PERSONAL COMPUTER
— Editor's Choice, May 1990

WHICH COMPUTER?—UK
— Best Buy Spreadsheets, October 1990

BITS & BYTES—NEW ZEALAND
— Editor's Choice, December 1989/January 1990 Special Edition
— Editor's Choice, February 1991

MIKRODATOM—SWEDEN
— Best Spreadsheet of the Year 1990

SYSTEMS INTEGRATION
— Applications Software Finalist, December 1990

GOVERNMENT COMPUTER NEWS
— Top-Rated Spreadsheet, June 1990

PC RESOURCE
— Best Numeric or Data Business Application

ELECTRONICS TEST
— Best Spreadsheet for Engineers and Scientists, July 1990

CPA MICRO REPORT
— 1990 CPA/MR All-Star Software, Best Spreadsheet for CPAs, July 1990

MANAGEMENT ACCOUNTING
— Best Spreadsheet Value, November 1990

THE CRITICS RAVE

"BEST BUY"

"Looking for everything Excel offers (without Windows' overhead), plus full 1-2-3 compatibility? Look no further than our Best Buy, *Quattro Pro*."
PC World—April 1990

"OVERALL MS-DOS SOFTWARE PRODUCT OF THE YEAR"

"In this year's Product of the Year ballot, InfoWorld readers chose many products that pushed the performance or technology envelope. . . . *Quattro Pro* proved that a spreadsheet could offer all the latest features and still run on an 8088-

based PC with only 512K of RAM. It includes innovative spreadsheet linking features, superior graphics, great output, and advanced database access, all organized by a modern and customizable user interface." InfoWorld—March 5,1990

"BEST SPREADSHEET SOFTWARE"

"*Quattro Pro* earned our highest score among high-end spreadsheets with it combination of Excel-like and 1-2-3-like features. It works smoothly even on 640K machines—and superbly on higher-end systems. With a splashy, graphical interface and extensive features set, this is a software package with a bullet." InfoWorld—March 5, 1990

"AWARD FOR TECHNICAL EXCELLENCE"

"Seeing *Quattro Pro* in action is believing: this is the top-dog spreadsheet. *Quattro Pro* clearly advances the state of the art in application software with its combination of performance, feature richness, and compatibility."
PC Magazine—January 16,1990

"ALL STAR SOFTWARE: BEST SPREADSHEET FOR CPAS"

"*Quattro Pro* pushes 1-2-3 out of the picture. It runs well on older computers that haven't had expensive memory implants and processor transplants. Recommend it . . . for clients who won't part with old favorites."
CPA Micro Report—July 1990

"BEST IN ITS CLASS"

"*Quattro Pro* combines many of the best elements of 1-2-3, Release 3 and Excel into one extremely powerful package, including impressive 3-D consolidation capabilities and state-of-the-art presentation-quality graphics. It's easy to use and nearly perfectly compatible with the 1-2-3 standard."
InfoWorld—January 22, 1990

"MOST VALUABLE PRODUCT OF 1989"

". . . the 640K PC has been temporarily revived by Borland International's *Quattro Pro*. . . . VROOMM (Virtual Runtime Object-Oriented Memory Management technology) allows *Quattro Pro* to work with larger spreadsheets than, say,

the memory-hungry Lotus 1-2-3 Release 3. . . . and *Quattro Pro* can handle spreadsheets up to three times larger than the largest worksheet Release 3 can manage."
PC/Computing—January 1990

"BEST SPREADSHEET FOR ENGINEERS AND SCIENTISTS"

"*Quattro Pro* is the 1-2-3 compatible spreadsheet that Lotus wishes it had introduced. It has the speed of version 2.2, the power of version 3.0, and dozens of features Lotus forgot to add to either version. Pro also has better graphics and hard copy than either Lotus program."
Electronics Test—July 1990

TOP-RATED SPREADSHEET

"*Quattro Pro* stands head and shoulders above the rest. In the latest poll, *Quattro Pro* left the competition behind in the compatibility, speed, efficient memory use and value. Overall it out-performs the others . . ."
Government Computer News—June 1900

"AWARD OF DISTINCTION"

"*Quattro Pro* is a joy to use. Its mouse support extends beyond simple pointing. The interface includes scroll bars and a user-programmable mouse palette . . . Once you've used a mouse for spreadsheet manipulation, you may never want to go back to your arrow keys."
Byte—February, 1990

"AWARD OF EXCELLENCE"

Quattro Pro provides more features and better performance than Lotus 1-2-3 at a lower price, and yet it still runs on any DOS machine. . . . Borland provides in Quattro Pro a group of features not found in any single version of the market leader 1-2-3."
Byte—January 1991.

HIGHEST SCORE IN SPREADSHEET REPORT CARD

"*Quattro Pro* demonstrates that innovation comes not only from new ideas, but also by combining existing concepts into a superior package. Borland International's entry into the

high-end market merges the best features of the two leading spreadsheets—and tosses in some improvements for good measure. . . . This spreadsheet may just be the best available in terms of performance, capacity, ease of use."
InfoWorld—November 13, 1989

"EDITOR'S CHOICE"

"When all is said and done, the best spreadsheets for graphics are the best ones for everything else. . . . Borland's Quattro Pro offers the strongest graphics of any character-based spreadsheet. . . . Quattro Pro provides a full set of charting options, drawing tools, a wealth of high-quality fonts, the ability to print graphs embedded in worksheets, and even a way to produce simple screen-show presentations—all in a package that works well under DOS on ordinary PCs. . . . If you need excellent graphics . . . this is the place to go."
PC Magazine—December, 1990

AWARD FOR TECHNICAL EXCELLENCE

"You get a better spread of functionality in *Quattro Pro* than you find in Lotus 1-2-3 release 2.2 or even release 3. *Quattro Pro* delivers a workable variety of three-dimensionality, a publishing capability and a whole lot of other goodies in a single package that benefits from high-spec hardware without absolutely requiring it."
PC Magazine—December 1989

BEST SPREADSHEET VALUE

". . . *Quattro Pro* is a powerful contender for best all-around spreadsheet, rivaling both Lotus 1-2-3, version 3, and Microsoft's Excel. . . . It's flexibility and power make it a superior choice for business, education, or personal use."
Management Accounting—November 1990

EXHIBIT 6

LIBRARY OF CONGRESS
COPYRIGHT OFFICE
Washington D.C. 20559

January 20, 1987

O'Sullivan Graev & Karabell
30 Rockefeller Plaza
New York, New York 10112

Attention: Kerry L. Konrad

RE: 1-2-3 SCREENS, RELEASE 1 AND 1-2-3 SCREENS, RELEASE 2 submitted on behalf of Lotus Development Corporation

Dear Mr. Konrad:

We have received your applications for registration of copyright claims in these works. In space 2 of the applications you describe the authorship as "Visual display/computer graphics." In space 3 of the applications you show that a previous registration has been made for each work and that the purpose of the present applications is "To obtain additional protection of VA copyright for visual element of work."

It is the position of the Copyright Office that textual screen displays embodied within the computer program that generates them are covered by the registration for the program, without either the need or justification for separate registration for the displays. Because the displays are considered to be an integral part of the program, the authorship in the displays appears to be the same as that contained in the program. Moreover, the Copyright Office would not register a claim in the format or layout.

We are, therefore, filing the applications without further action. The fees are non-refundable.

Sincerely yours,

/s/ JULIA B. HUFF
Julia B. Huff
Supervisory Copyright Examiner
Visual Arts Section
Examining Division

EXHIBIT 9

LOTUS™ 123

USER'S MANUAL

For the IBM® Personal Computer and the COMPAQ™ Portable Computer

* * *

[60] COMMANDS

Using Menus

One of the cardinal principles of the computer world is:
The more characters you type, the greater your chances of making a mistake.

And one of the main questions asked by users of computer programs is:
What do I do next?

In recognition of both these points, 1-2-3 relies heavily on **menus** to handle its communication with you. A menu is simply a list displayed in the control panel that informs you of your options and lets you select one of them. Menus play a role in several 1-2-3 command situations (figure 33).

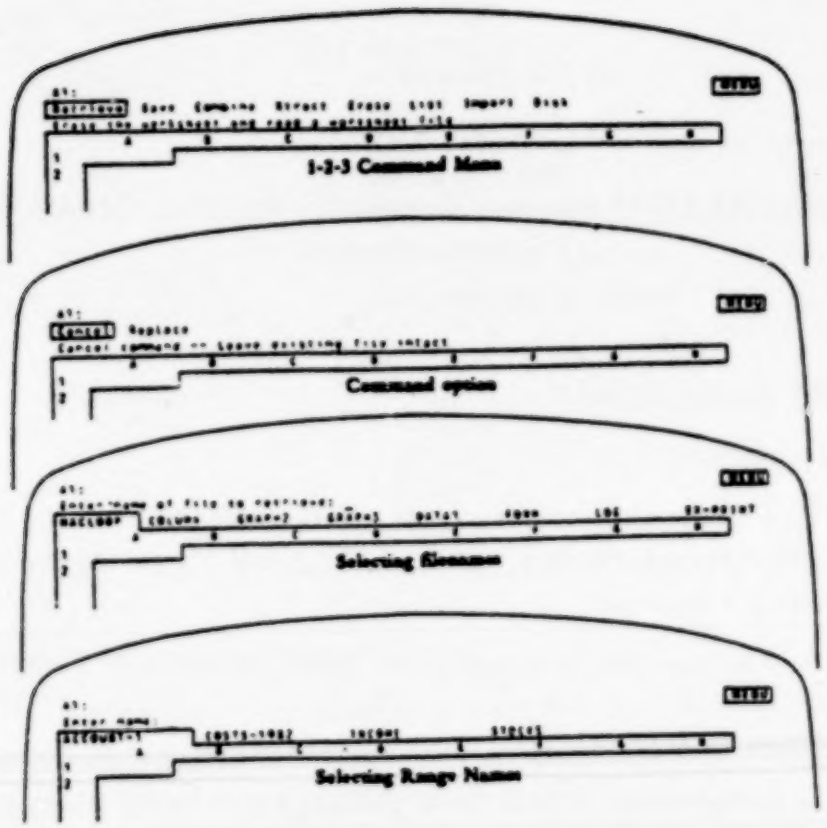


Figure 33. **Types of Menus**

In some cases, the list of options is complete. When 1-2-3 shows you a list of command keywords, for example, that's all there is. You must select one of these keywords in order to proceed with the command.

EXHIBIT 14

CONFIDENTIAL

INTEROPERABILITY QUALITATIVE STUDY

August, 1988

MRD #212

Prepared by:

Linda Christle
Marketing Research Department

In Collaboration With:

Sunday Lewis
Techvantage, Inc.

Research conducted:

August, 1988

[iv] *SUMMARY OF FINDINGS*

Participants

— It was difficult to find MIS/InfoCenter Managers who supported both IBM PC and Macintosh environments.

— End user participants found the subject area extremely difficult to discuss.

Environmental Issues

— In those environments where PCs and Macs exist, both platforms will grow in use. Those shops that are predominantly one platform or another have a tendency to want to maintain one predominant platform.

— The majority of firms have software standards lists which have been relaxed in recent months because of user demands. 1-2-3 and Mac Excel are the predominant spreadsheets; however, PC Excel is gradually being accepted as an alternative standard for some applications.

— OS/2 and PM hold promise but most participants will remain skeptical until application software that clearly demonstrates the benefits in switching is available.

— Mac users are considered different from the rest of an organization. Either they are in marketing, where desktop publishing is critical, or they are "academics."

User Interfaces

— Interoperability is not a commonly used term. However, as the discussion unfolded, participants redefined interoperability to mean cross platform data/file sharing and macro compatibility, and (to a much lesser extent) cross platform user interface.

— "1-2-3-ness" is not as easily identified as "Mac-ness." When probed, 1-2-3 meant a structured command set, row and column grids, and (/). Mac features were more easily defined: icon-driven, easy to use, mouse-driven, outstanding graphics and a consistent user interface across different software products.

— The 1-2-3 user interface is considered familiar and functional—a "workhorse." Most considered it the best UI at the time of its announcement, but less so today. It was not considered elegant.

* * *

[4] "The Macs are used by Marketing people who crank out brochures and a lot of graphics and by our Planning group who is constantly cranking out proposals . . ."

"We use ours for desktop publishing."

"Almost all of ours are used for Desktop Publishing."

"Info Resources."

"Budgeting, planning, presentations and in-house SW development."

Mac users fall into definite categories.

Attendees reported a definite difference between Mac and PC users. However, the difference was not so much in job responsibility but in personality types and attitudes . . .

"Academic types use Macs."

"Mac people are the sweater and jeans type rather than the 3-piece suit type."

"Our Mac users tend to be very demanding of what our systems can give them, much more so than IBM users."

"Mac users are very loyal. More than likely they picked the MAC, it wasn't given to them. More than likely they're hackers."

PC users overwhelmingly use 1-2-3.

The vast majority of PC users had chosen 1-2-3 as their spreadsheet. That decision was made when 1-2-3 first came out and was obviously far superior to VisiCalc. This strong base perpetuated sales/use of 1-2-3 and is the reason for 1-2-3's success today . . .

"Everyone I work with uses 1-2-3."

"When 1-2-3 came out you thought it was great because there was nothing to compare it to but VisiCalc."

"When it first came out it was like WOW!, this is pretty neat."

"There's a certain amount of momentum there because people already know how to use 1-2-3 and it does the job for them. These people don't even look to switch. There is also

a tremendous momentum to keep 1-2-3 so as not to have to reinvest in training. That's one of the things that keeps products where they are—the investment in training."

* * *

[14] "The concept (of a language for experienced application developers) is not appealing. Lotus should be more for the end-user."

The common interface is somewhat important but data/macro compatibility is more important.

The consensus here is that data/macro capability is the most important item. Attendees were willing to give up a little on the UI for the data/macro compatibility . . .

"All Lotus products should have data/macro compatability. That's real important and the UI may have to be sacrificed a little for that."

"Compatibility is the most important thing. If the UI is so important to you—you buy a Mac."

"We have an awful lot of remote users that feed in financial information. We can't expect them all to go up to OS/2, so there are still going to be old PCs. We can't ask the remote users to make a commitment to new hardware and new software. So, they're going to have to have data sharing and macro compatibility. I want data sharing and macro compatibility over everything else."

". . . if the data is incompatible, then as far as I'm concerned, it's useless. If I have to go changing everything; we may have 2,000-3,000 programs floating around. If they all have to be changed, then I'm not going to buy the product."

The users are comfortable with 1-2-3's UI and are pleased with its ability to help them do their job.

On the whole, 1-2-3 users related well to the 1-2-3 UI. They are comfortable with it from so many years of use and appreciate the fact that 1-2-3 helps them get out the work when they really need to. Although praised for its functionality, they agreed that the UI was no longer "elegant."

"At one time it was [elegant], but not anymore. I think when it first came out it was like WOW!, this is pretty neat."

"One of the great things that's always been about Lotus is that it's a 'workhorse.' It ain't pretty but it does the same thing everyday. It does the job everyday with a consistency across generations. Everybody's starting to add bells & whistles to spreadsheets making them something they're not. I think the vendor community has to start

* * *

[15] *Each platform's spreadsheet should be optimized for its own environment.*

Each platform's spreadsheet product should contain all of the features necessary to make it competitive in its own environment. However, a core set of features across platforms should be retained.

"There should be a baseline, a core set of functions across all. Those would be the most used functions. Extensions are something we all live with."

"I'm not desperately worried about every function. I'm worried about core functions working, /FILE RETRIEVE working on both of them. But, if there are special functions that are key to my VM environment, I'm not worried that they should be exactly duplicated."

"The old functions have to be your new base functions. Then you build on them, so that your old program works on the new system. That has to be."

Attendees are in agreement that macro compatibility is KEY.

Participants felt it was important to preserve the time spent developing macros. They considered macro compatibility critical, no matter what version they were working with . . .

> "The key is to be able to bring what we've already got forward, even though it may look a little different. The data and macros should come forward so that the users can be operational within minutes of installing."
>
> "If you're going to move programs and data back and forth you also want to move the whole spreadsheet back and forth which would include any macros you had."
>
> "We expect macro compatibility."

EXHIBIT 15

Europäisches Patentamt
European Patent Office
Office européen des brevets

Publication number: 0 428 261 A2

EUROPEAN PATENT APPLICATION

Application number: 90310897.5

Date of filing: 04.10.90

Int. Cl.[5]: GO6F 3/033

Priority: 13.11.89 US 436517

Date of publication of application:
22.05.91 Bulletin 91/21

Designated Contracting States:
AT BE CH DE DK ES FR GB GR IT LI LU NL SE

Applicant: LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142(US)

Inventor: Wolf, Richard J.
10 Everett Street
Sherborn, Massachusetts 01770(US)

Inventor: Fossgreen, Donald J.
675 Millbury Street
Worcester, Massachusetts 01607(US)

Inventor: Rosen, Robert
588 Huron Avenue
Cambridge, Massachusetts 02138(US)

Inventor: Vento, Christopher M.
353 Pleasant Street
North Andover, Massachusetts 01845(US)

Inventor: Papanek, Doris Wells
510 Santa Margaritad
Menlo Park, California 94025(US)

Representative:
Blatchford, William Michael et al
Withers & Rogers
4 Dyer's Buildings Holborn
London EC1N 2JT(GB)

Computer user interface.

A computer user interface that comprises a system and methods to provide a control hierarchy, control marking, a hybrid gammar, and resized and/or repositioned dialog boxes. Control hierarchy divides controls, assigns each control a label, displays the labels according to a design hierarchy, and operates on the controls according to a functional hierarchy. Control marking distinguishes controls, vis addition of graphic elements or changes in labels—referred to as glyphs. Hybrid grammar simultaneously allows a user to specify operations and data objects in a noun verb relationship or verb noun relationship. Finally, resized and/or repositioned dialog boxes enable a user to select a control that requires selection of data objects that are possibly obscured by the dialog box.

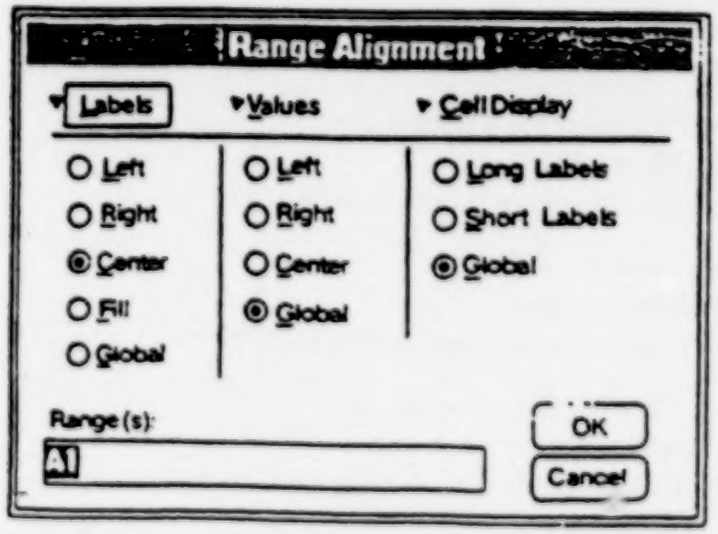


FIG 1

* * *

[9] Before and after the messages reach or are sent to the application 14, however, the HDBM filters the messages from the OS/WM 12 so that both the application 14 and the OS/WM 12 receive only relevant messages. For example, HDBM might use an initialization message sent from the OS/WM 12 to autoposition the dialog window to an area of the screen when the window becomes visible.

The HDBM is an extension of the window manager in OS/WM 12. The window manager, e.g., presentation Manager®, has its own dialog box manager. However, HDBM supports the hierarchical keystroke mnemonics described above in connection with Control Hierarchy and Control Marking. In addition, the HDBM provides access to a group of control windows contained within the dialog box that are not part of sthe hierarchy, e.g., "OK" and "Cancel" controls and fully maintains the standard presentation managerc mouse selection and pointing conventions for dialog boxes.

Referring again to Fig. 6, note that the HDBM is divided into two parts, namely the Front End Filter 18 and the Back End Filter 20. The Front End Filter 18 filters the messages from the OS/WM 12, routes or translates if necessary the relevant messages for the application 14, and then passes the messages to the application.

At run-time, the HDBM dialog box loader 13 captures and uses the parent/child arrangement of the controls specified in the definitions file 16 to graph the hierarchy of the dialog box so the HDBM can "navigate" between controls, i.e., move from one group of controls to another in proper sequence and move from one control to another within a group. Once the HDBM dialog box loader 13 has captured the hierarchical relationship between the controls, it "flattens" the dialog box window by making all the controls within the window children of the dialog window. This is done to simplify the message protocol imposed by the window manager 12 and controlled by the application 14 when creating and/or processing the dialog box.

Next, the application 14 filters and translates the messages it receives from the Front End Filter 18. The application 14, based on the definitions of the dialog box window and control windows in the definition file 16, executes any application specific operations on the affected control window and dialog box window. For example, the application 14 might validate text in a edit control box.

Next, the application 14 passes any messages that are not relevant, e.g., OS/WM primitive operations to the Back End Filter 20. The Back End Filter 20 filters the messages and executes hierarchical operations. For example, if the selection of one control makes it impossible to select a second **[10]** control, the Back End Filter 20 removes a glyph from the second control. Finally, the Back End Filter 20 passes the messages to the OS/WM 12 for any default processing that may be needed.

In addition, the HDBM fields all mouse events and keystroke events while the dialog box is active. If a control "windowthat supports selection outside the dialog box has the input focus, a mouse event or certain keystroke events cause the dialog box to be resized and/or repositioned as described above, when the events indicating that the selection has been completed are received by the HDBM, the dialog box is restored to its original size and position. The resizing and repositioning is a low priority activity so that if the events indicating the selection have been completed are received prior to the onset of the resizing and repositioning of the dialog box, the resizing and repositioning do not take place.

Further, in connection with the noun verb and verb noun grammar, the application main body 11 supplies the dialog box with the object that the user has selected prior to issuing the command that invoked the dialog box so that the edit control box specifying the selection can be initialized accordingly with the current object.

Other embodiments are within the following claims.

Claims

1. A method of providing a plurality of controls in a software application comprising the steps of:

dividing said controls into functional hierarchical groups;

assigning labels to each of said controls and each of said groups;

displaying said labels according to a design hierarchy; and

operating on said controls according to said functional hierarchical groups.

2. A method of displaying a plurality of controls in a software application some of which are available comprising the steps of:

determining which of said controls are available at particular time;

generating a glyph for each of said available controls at said time; and

displaying one of said glyphs in proximity to each of said available controls at said time.

3. A method of displaying a plurality of controls in a dialog box some of which are available comprising the steps of:

determining which of said controls are available at a particular time:

generating a glyph for each of said available controls at said time; and

* * *

EXHIBIT 16

United States Patent

Lyons et al.

Patent Number: 4,989,141

Date of Patent: Jan. 29, 1991

COMPUTER SYSTEM FOR FINANCIAL ANALYSES AND REPORTING

Inventors: Richard J. Lyons, West Hempstead; Kevin F. Nolan, Middle Village, both of N.Y.; Wah C. Chu, Fairfield, Conn.

Assignee: Corporate Class Software, Stamford, Conn.

Appl. No.: 55,823

Filed: Jun. 1, 1987

Int. Cl.[5] G06F 15/30

U.S. Cl. 364/408

Field of Search 364/408

References Cited

U.S. PATENT DOCUMENTS

4,321,672 3/1982 Braun 364/408

OTHER PUBLICATIONS

"Using Spreadsheets to Monitor Your Portfolio", Thomas A. Meyers, PC Magazine, Apr. 15, 1986.

Dow Jones Software, product brochure, Spreadsheet Analysis.

Teddi Converse, *Reflex In Business*, pp. 204-217, 1985.

Primary Examiner—Allen MacDonald

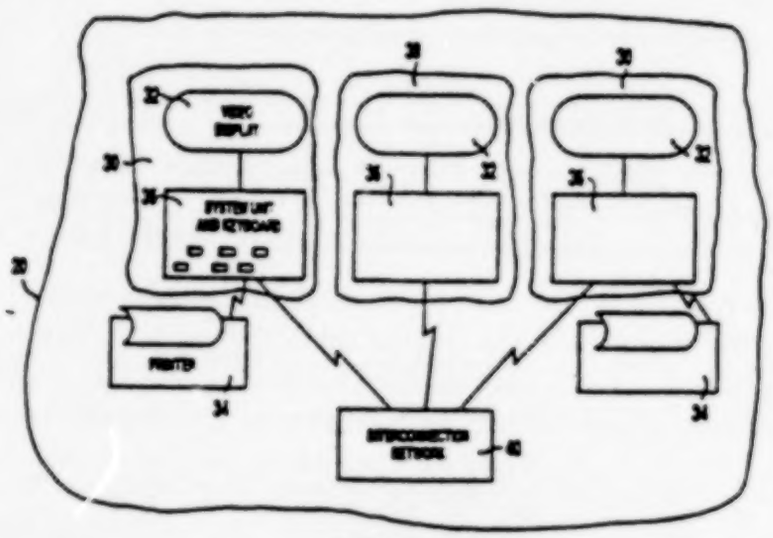
Assistant Examiner—Gail Hayes

Attorney, Agent, or Firm—Pennie & Edmonds

ABSTRACT

An advanced financial reporting and analysis software package is described. The package collects, organizes, manages and consolidates financial data and provides user defined capabilities for creating financial and corporate reports. Financial data is organized into four business classifications or dimensions: Schedule, Entity, Period and Type. Data is stored in the system in such a way that all data associated with a particular Schedule, Entity, Period and Type is identified by that particular SEPT value. To accommodate automatic data entry, a mapping means or template is provided that specifies for each different input spreadsheet the location of the first data cell in the spreadsheet and the size of the spreadsheet. Data is read from the data store by various report and spreadsheet generating functions which convert data associated with particular SEPT values to desired output formats.

10 Claims, 29 Drawing Sheets



COMPUTER SYSTEM

* * *

TABLE VII

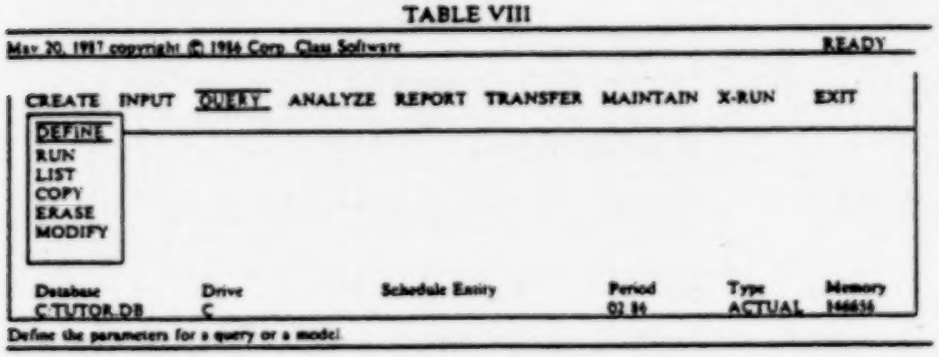
07:06:38 May 19, 1987 QUERY Pg 1

	FY 87 ABC ACTUAL A	FY 87 ABC BUDGET B	FY 87 ASC ACTUAL C	FY 87 ASC BUDGET D
1 Sales	161	170	249	274
2 Expenses	178	163	163	174

As will be apparent the report is a two-dimensional array of data. The column headings in this example specify Periods, Entities and Types. Other combinations of up to three of the four SEPT dimensions are possible. These headings are selected by the user by means of the QUERY function and a sub-function named ACROSS. In this example, the row headings are elements of a Schedule; but other headings selected from the SEPT dimensions not used as column headings can be used. The headings are specified by the user by means of QUERY and a sub-function named DOWN.

Table VIII is illustrative of the screen that is displayed when the QUERY function is selected.

TABLE VIII



The DEFINE function allows the user to define a query. For example, the user will set up column and row headings for the report, set up selection criteria, set up format and specify output mode. Column and row headings will be selected from available values of the four SEPT categories.

The RUN function generates a report or a worksheet in response to a query defined by the DEFINE function.

The LIST function lists the available queries that have been established by the DEFINE function. This will allow the user to review his selection criteria, report format and output modes.

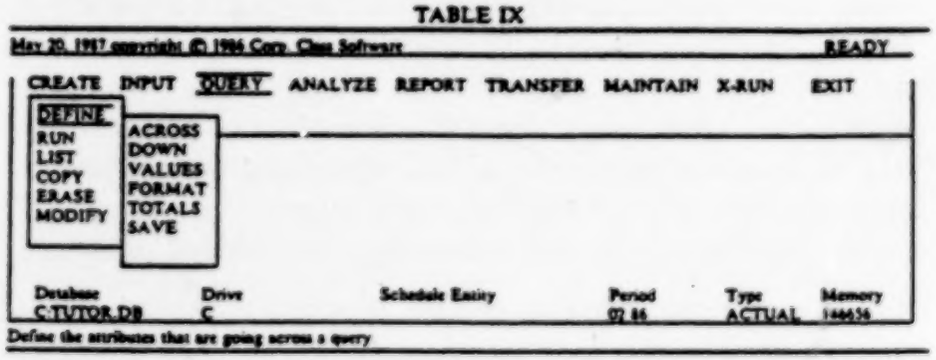
The COPY function allows the user to copy an existing query to a new query.

The ERASE function deletes an existing query.

The MODIFY function allows value modifications to an existing query.

By selecting the DEFINE function, the user prompts the system to display a submenu which lists all subfunctions available from the DEFINE function. Table IX is illustrative of the screen displayed to the user once he has chosen the DEFINE function. The user may now go through all of these subfunctions in order to define his query.

TABLE IX



The ACROSS subfunction allows the user to select (at most three of the four SEPT variables) (SCHEDULE, ENTITY, PERIOD, TYPE) to be used to define the column headings for a query. The order of selection of the attributes determines their relative levels in the column heading.

The DOWN subfunction allows the user to select the remaining SEPT variables (at most three of the four SEPT variables) to be used to define the row headings for a query. The order of selection of the attributes determines their relative levels in the row headings.

The VALUES subfunction permits the user to specify available range values for use as row headings.

Once all values are selected, the user may define his report format by choosing the FORMAT subfunction from the Query/Define submenu illustrated in Table IX. This subfunction generates a menu that leads the user through specification of headers (report titles), trailers (footnotes), and data precision (the number of decimal places) and provides the user a choice between sending the report to a printer or an electronic work-

* * *

EXHIBIT 19

LIBRARY OF CONGRESS
Washington D.C. 20559

June 15, 1989

Fenwick, Davis & West
Attn: Stuart P. Meyer, Esq.
2 Palo Alto Square, Suite 800
Palo Alto, CA 94306

Re: SPACEMAN II and SPACEMAN III

Dear Mr. Meyer:

Registration was completed for both works and certificates of registration forwarded to you earlier. We apologize for the delay in writing on this material.

During the past year, the Copyright Office reviewed its registration and deposit practices for computer screen displays. As a result of this review the Office announced its decision on June 10, 1988 to allow only a single registration for all copyrightable expression embodied in a computer program, including computer screen displays, and owned by the same claimant. See the enclosed leaflet for additional information.

As the leaflet explains, where the application specifically refers to screens, or where screens are deposited whether mentioned on the application or not, those screens will be examined for copyrightability. Both of these computer programs include a number of screen displays on videotape.

Copyright protects "original works of authorship fixed in any tangible medium of expression." To be regarded as an "original work of authorship," a work must contain at least a minimum amount of original artistic, literary, or musical material. In all cases, only the expression of the author is protected by copyright. The ideas that are embodied in the author's work are not copyrightable.

The Office has a well-established practice of refusing to register claims to copyright in mere format of text or images; this practice is based on the statutory prohibition against copyright in ideas, systems, concepts, or discoveries. 17 U.S.C. 102(b). Similarly, copyright cannot be secured for names, titles, and phrases such as column headings, simple checklists, and the like, nor can it be secured for the format, arrangement, or typography of a blank form or similar work. Thus, in general, menu screens and similar functional interfaces consisting of words or brief phrases in a particular format are not registrable.

Our examination of the videotapes of each work revealed some menu screens which we would not consider registrable. Other screens contained copyrightable material. Registration was completed and, of course, the entire videotape is part of the deposit since it was submitted as a collection. We point out, however, that not all of the individual screens would be registrable if submitted separately.

This letter is for your information only. No reply is necessary.

Sincerely yours,

/s/ PHILL L. GILL
Phill L. Gill
Head, Literary Section II
Examining Division

Enclosures:
Screen display leaflet

EXHIBIT 20

COMPUTER PROGRAMS AND RELATED SCREEN DISPLAYS

The Copyright Office has registered claims to copyright in computer programs as literary works since 1964, provided that the work contained at least a minimum amount of original literary authorship. The Office presumed that these registrations extended to all of the copyrightable content in the computer programs. Recently the practice of making separate registrations for computer screen displays raised a number of questions. The courts have affirmed in several videogame cases that pictorial and graphic screen displays may be separately registered. However, courts offered disparate opinions on protection for screen displays other than videogame displays.

To elicit comments on the registration and deposit of screen displays, the Copyright Office held a public hearing in September 1987, and accepted further written comments through October 9, 1987. As a result of that review, in June 1988 the Office announced a decision to require that all copyrightable expression embodied in a computer program owned by the same claimant, including computer screen displays, be registered on a single application form.

This decision also applies to videogame displays; these claims will be treated the same as other claims that include authorship in a computer program and screen displays. A single registration will be made for the computer program and any related audiovisual authorship owned by the same claimant. Separate registrations will not be made.

The Copyright Office has consistently held the view that a single registration is sufficient to protect the copyright in a computer program, including related screen displays, without a separate registration for screen displays or reference to the displays in the "nature of authorship" description on the application. An application may give a general description such as "entire work" or "computer program." This descrip-

tion would cover any copyrightable authorship contained in the computer program and screen displays, regardless of whether identifying material for the screens is deposited.

Applicants who previously made a single registration for a computer program should be assured that the registration covers all of the copyrightable content of the computer program. The Office will not make a new basic registration or a supplementary registration to allow a separate claim in the screen displays. Neither will the Office accept identifying material for the screens contained in any previously registered computer programs.

HOW TO REGISTER COMPUTER PROGRAMS/SCREEN DISPLAYS

A single registration will be made in the class in which authorship predominates. Because the computer program is a literary work, literary authorship will predominate in most works, including many in which there are screen graphics, and registration will generally be appropriate on Form TX. If pictorial or graphic authorship predominates, registration may be made on Form PA as an audiovisual work or on Form VA.

The registration will extend to any related copyrightable screens, regardless of whether identifying material for the screens is deposited. **Where identifying material is deposited, it will be examined for copyrightability.**

Where the applicant refers to screen displays on the application, identifying material for the screens *must* be deposited.

To make registration for a computer program and its related screen displays:

* Select application form according to which kind of authorship predominates.
* Follow the chart on the next page to complete space 2 of the application and to determine whether to file identifying material *for the screen displays*.

* Fill out and submit the application along with:
  * The appropriate deposit for the *computer program* (preferably the first and last 25 pages of the program in source code);
  * A non-refundable filing fee of $10;
  * Optional identifying material for the screen displays.

How To Complete The Application	What To Deposit
Space 2	*ID Material for Screens*
Opt. 1 "Entire work" or "Entire computer program"	You may choose NOT to deposit ID material
Opt. 2 "Entire work" or "Entire computer program"	You may choose to deposit ID material
Opt. 3 "Entire computer program including text of screen displays" or "Entire computer program including audiovisual material" or "Entire computer program including artwork on screen displays"	You MUST deposit ID material
Note: The description of authorship on the application should NOT refer to elements such as "menu screens," "structure, sequence and organization," "layout," "format" or the like.	Note: Identifying material for screen displays should consist of printouts, photographs, or drawings clearly revealing the screens. For works that are predominately audiovisual, such as video-

games, 1/2 inch VHS videotape is acceptable. ALL screen identifying material will be examined for copyrightability.

EXHIBIT 21

UNITED STATES COURT OF APPEALS
FOR THE NINTH CIRCUIT

No. 89-15683

ASHTON-TATE CORPORATION, a Delaware corporation,

Appellee/Plaintiff,

vs.

RICHARD ROSS; BRAVO TECHNOLOGIES, INC.,
a California corporation,

Appellants/Defendants.

RICHARD A. ROSS and BRAVO TECHNOLOGIES, INC.

Counterclaimants-Appellant,

vs.

ASHTON-TATE CORPORATION,
and DOES ONE through FIVE,

Counterdefendants-Appellees.

On Appeal from the Order of the United States
District Court for the Northern District of California
No. C-88-2844 DLJ

Honorable D. Lowell Jensen, District Judge

PAUL N. MCCLOSKEY, JR.
MCCLOSKEY & KAYS
3000 Sand Hill Road
Building One, Suite 235
Menlo Park, California 94025
Telephone (415) 854-7770

Attorneys for Appellants/Defendants
RICHARD A. ROSS and
BRAVO TECHNOLOGIES, INC.

---

[25] *authorship* that went beyond the *ideas* expressed in the status screen. According to the Northern District of Georgia:

> In the instant case, however, the arrangement of the status screen involves considerable stylistic creativity and authorship above and beyond the ideas embodied in the status screen. It cannot be said that the idea of the status screen . . . could not have been expressed in a large variety of ways. The defendants have never contended that they could not have arranged the parameters/commands in a wide variety of patterns without hampering the operation of their program. The parameters/commands could have been arranged and delineated in an almost infinite number of horizontal and vertical patterns and groupings that would be substantially dissimilar to the arrangement and groupings utilized by the plaintiff. Likewise, the defendants could have used a wide variety of techniques to indicate which symbols the user should type to effectuate a command. . . . The modes of expression chosen by the plaintiff for its status screen are clearly not necessary to the idea of the status screen. Therefore, the plaintiff's mode of expression of the status screen does not merge with the idea of the status screen.

Softklone, 659 F.Supp. at 460. Exhibits 8 and 9 display numerous decisions by the authors about the ordering of the commands and their arrangment in the user interface. The fact that the authors of these design documents chose the order and groupings displayed, out of a nearly infinite number of possibilities, constitutes creative authorship according to the Softklone court.

The District Court in the present case erroneously found that Exhibit 8 was not copyrightable because it did not contain source code. (Excerpts, Tab 103, p. 10). In *Softklone*, the Northern District of Georgia stated that a status screen constitutes a "literary work," within the meaning of 17 U.S.C. § 101, when written on a piece of paper, such as in Exhibits 8

* * *

EXHIBIT 23

United States Patent

Crehan et al.

Patent Number: 4,611,306

Dated of Patent: Sep. 9, 1986

DISPLAY MENU/CHART KEY

Inventors: Donald T. Crehan, Gaithersburg, Md.; Charles J. Lovell, Austin, Tex.

Assignee: International Business Machines Corp., Armonk, N.Y.

Appl. No.: 493,572

Filed: May 11, 1983

Int. Cl.[4] G06F 15/62; GO6F 3/037; GO6F 15/626; GO9G 3/02

U.S. Cl. 364/900; 364/521; 340/712; 340/747

Field of Search ..... 364/200 MS File, 900 MS File, 364/521, 522; 340/712, 721, 747

References Cited

U.S. PATENT DOCUMENTS

3,879,722	4/1975	Knowlton	340/705
4,430,725	2/1984	Elliot et al.	364/900
4,435,772	3/1984	Suzuki et al.	364/520
4,441,163	4/1984	Leikam et al.	364/900
4,454,576	6/1984	McInroy et al.	364/200
4,456,969	6/1984	Herzik et al.	364/900
4,458,311	7/1984	Clements et al.	364/200
4,458,331	7/1984	Amezcua et al.	364/900
4,476,462	10/1984	Feldman	340/711
4,479,197	10/1984	Haag et al.	364/900

FOREIGN PATENT DOCUMENTS

8200726	3/1982	PCT Int'l Appl.	340/712

OTHER PUBLICATIONS

"Lotus Development Corporation's 1-2-3" by Gregg Williams; Byte, Dec. 1982, pp. 182-198.

"Introduction 1-2-3 It'll Have Your IBM/PC Jumping Through Hoops"; Byte, Apr. 1983, vol. 8, No. 4.

"Lotus 1-2-3" advertisement; Byte, Feb. 1983, vol. 8, No. 2.

Xerox Disclosure Journal, vol. 4, No. 5, Sep./Oct. 1979, p. 641, H. Wallace Swanstrom: "Selection of Options".

IBM Technical Disclosure Bulletin, vol. 25, No. 3B, Aug. 1982, pp. 1785-1786, Adam et al.: "Menu and Command Language Interface for Information Processing System".

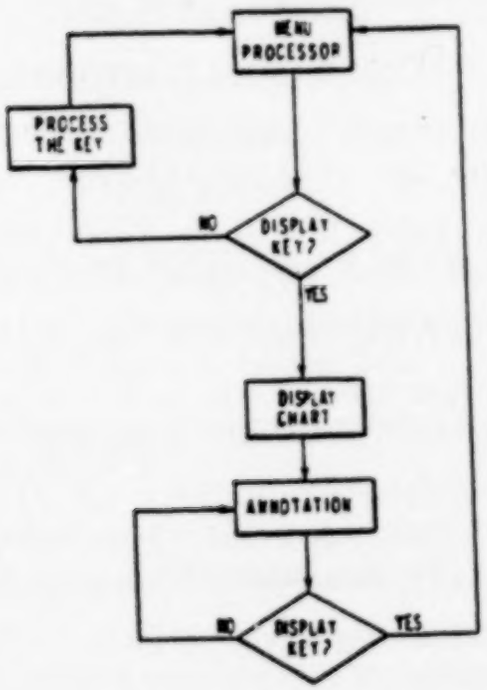
Wescon Technical Paper, vol. 26, Sep. 1982, pp. 32/2/1-32/2/2, G. Romans: "Color Program Without Keyboards".

Primary Examiner—James D. Thomas
Assistant Examiner—Thomas Lee
Attorney, Agent, or Firm—C. Lamont Whitham

ABSTRACT

A display menu/chart key is provided in a word processing system to allow a system operator to toggle back and forth between the display of a menu used to define a graphics image and the graphics image to the extent that the graphics image has been defined by the menu. This provides an easy interface that allows the operator to visually check and revise the definition of a graphics image such as a bar, line or pie chart. By using this key, the operator can quickly see how changing a menu item will cause the chart to be changed and the particular portion of the chart that will change. When the key is pressed while a chart is being displayed, the last menu being processed will again be displayed.

3 Claims, 3 Drawing Figures



* * *

Suppose that a bar chart is not selected but instead a line chart is selected. In this case, the following menu is displayed:

LINE CHART DESCRIPTION

Id.	*Item*	*Your choice*	*Possible choices*
a	Display Example Screen		
b	Number of Lines	1	(1-4)
	Vertical Axis		
c	Total Number of Labels	5	(2-15)
d	Top Label Value	100	
e	Bottom Label Value	0	
	Horizontal Axis		
f	Total Number of Labels	3	(1-24)

When finished with this menu, press ENTER
Type ID number to choose ITEM, press ENTER:

Once the line chart had been defined using this menu, the "Data Entry" menu for the line chart would be displayed. This menu is quite similar to that shown above for the vertical bar chart except that the word "line" is substituted for the word "bar" at the heading of each of the four data entry columns.

Finally, suppose that a pie chart had been selected instead of the bar chart as illustrated in the "Create or Revise Chart" menu shown above. In this case, a simple two column "Data Entry" menu for the pie chart is displayed. The first column has the heading "Section Labels" and the second column has the heading "Values". FIG. 2 does show yet another possibility, that of a free-form chart, but that does not involve data entry. FIG. 2 is itself a good example of a free-form chart.

In the creation or revision of a chart as described above, it will be recalled that as each menu in succession is finished, the "Enter" key is pressed. Thus, by pressing the "Enter" key, the operator can page from the "root" to the "leaves" in the tree structure shown in FIG. 2. It is also possible to page backward from the "leaves" to the "root" by pressing another key provided specifically for that purpose. The problem solved by this invention is to provide an easy to use interface to the operator that gives the operator the capability to check the progress of the chart definition and to return to the current place in the chart definition with a minimum of keystrokes. Requiring a single keystroke was chosen as the optimum. Allowing no keystrokes suggests a timed image display, and this could be an undesirable interface in that multiple keystrokes would be required to display the chart image for long periods of time. Allowing, or, worse yet, requiring the operator to set or define the time intervals would make the interface more cumbersome. Further, more than a single keystroke would not satisfy the "minimum" requirement implicit in the easy to use interface.

The solution according to the invention is to provide a single key on the keyboard that causes the chart to be displayed

and also causes a return from displaying the chart to the menu in the chart definition process that was last active before the key was pressed to display the chart. For example, suppose the operator has elected to display a vertical bar chart but has not yet defined the axis parameters or entered any numerical data. By pressing the display menu/chart key, the operator can have the chart displayed which at this point would consist of horizontal and vertical axes. By again pressing the display menu/chart key, the last active menu would be displayed which in this case would be the "Axis Definition" menu. When the operator is finished with this menu the "Data Entry" menu will be displayed, but before entering data, the operator can again press the display menu/chart key to observe the axes of the chart has it has been defined. Pressing the display menu/chart key again will cause the "Data Entry" menu to be displayed again. At any time while the operator is entering data into this menu, the display menu/chart key can be pressed to see how the chart is developing. In this way, the operator has a simple way to check the progress of defining the chart by viewing it as it is currently defined, as well as having a simple way to continue defining the chart, or to modify some definition that had been previously entered, without having to page back and forth through several menus, or worse, having to start at the beginning of the process.

FIG. 3 shows a flow diagram illustrating the operation using the display menu/chart key according to the invention. The operator begins by defining the chart using the various menus that define the chart type, scaling, orientation, data to be plotted and so forth. During this time, the keyboard is monitored to determine if the display menu/chart key has been pressed. If it has not been pressed, the key that has been pressed is processed in the usual way. At some point during the definition, the operator may decide to see what the chart looks like at the current stage of the definition and so presses the display menu/chart key. This is detected by the menu processing software which then calls the display chart processing software. In response to this call, the chart is built and displayed to the extent possible with the current chart definition. At this point, the operator would be allowed to "annotate" the chart as may be desired. Application Ser. No. 493,581 entitled "Independent Image and Annotation Overlay with Highlighting of Overlay Conflicts" describes this process. After making an annotation or not, the operator might then decide to continue defining the chart, or to redefine some part or all of the chart. To do this, the operator again presses the display menu/chart key, and the display chart processing software recognizes this as the signal to return to the menu processing software. Those skilled in the art will understand that the nature of the "call" and "return" functions in software is that the point of operation will continue after a "return" at the point immediately after the "call". The procedure is well known and will not be described in any further detail. The important thing to understand is how the procedure is used in the operation of the invention. Thus, when the menu is displayed again, the operator can continue with the chart definition until completed or until s/he wants to view the chart again at the new level of completion.

The display menu/chart key according to the invention need not be an added key to the word processing system keyboard. In the preferred embodiment, the display menu/chart key is a software defined key that is recognized by both the menu processing software and the display chart processing software. In this way, the key may be used to toggle back and forth between the display of the current menu and the chart.

We claim:

1. A process for interactively constructing a graphic image on a video display terminal provided with a display menu/chart key for toggling between the display of a menu and the display of a chart comprising the steps of:

displaying to an operator a construction menu listing a plurality of selectable construction parameters which define the graphic image to be constructed;

displaying a data entry menu to said operator after the construction parameters have been selected in said construction menu and allowing the operator to enter data

or change data already entered in said data entry menu at any time during said process for interactively constructing a graphic image when said data entry menu is displayed;

upon detecting the actuation of said display menu/chart key while displaying said data entry menu, displaying to the operator a graphic image on said video display terminal as defined by entered and changed data at the time said display menu/chart key is activated; and

upon detecting the actuation of said display menu/chart key while displaying said graphic image, redisplaying said data entry menu so that the operator can view the progress of constructing the graphic image on said video display terminal as data is entered or changed in said data entry menu at any time during said process of interactively constructing a graphic image by toggling between the display of said data entry menu and the chart

2. The process for interactively constructing a graphic image on a video display terminal as recited in claim 1 wherein the operator is allowed to select construction parameters of change construction parameters already selected in said construction menu at any time during said process for interactively constructing a graphic image when said construction menu is displayed further comprising the steps of:

upon detecting the activation of said display menu/chart key while displaying said construction menu, displaying to the operator a graphic image on said video display terminal as defined by the construction parameters selected or changed when said display menu/ chart key was activated; and

upon detecting the actuation of said display menu/chart key while displaying said graphic image, redisplaying said construction menu so that the operator can view the chart as defined by construction parameters which have been selected and changed in said construction menu at

any time during the process of defining the graphic image to be constructed.

3. A process for interactively constructing a business chart on a video display terminal provided with a display menu/chart key for toggling between the display of a menu and the display of a business chart, the type of business chart being a bar chart, a pie chart or a line graph, said process comprising the steps of:

displaying to an operator a construction menu listing a plurality of selectable construction parameters which define the business chart, said construction parameters including the type of business chart and its scale, and allowing the operator to select construction parameters or change construction parameters already selected in said construction menu at any time during said process for interactively constructing a graphic image when said construction menu is displayed;

upon detecting the activation of said display menu/chart key while displaying said construction menu, displaying to the operator a graphic image on said video display terminal as defined by the construction parameters selected when said display menu/chart key was activated provided the type of business chart has been selected; and

upon detecting the activation of said display menu/chart key while displaying said graphic image, redisplaying said construction menu so that the operator can view the chart as defined by construction parameters which have been selected and changed in said construction menu at any time during the process of defining the graphic image to be constructed.

* * *

EXHIBIT 27

QUATTRO PRO DISPLACEMENT STUDY

December 17, 1990

MRD # 341

Prepared by: Gasteiger & Associates
in collaboration with Nancy Merz

* * *

[3] Methodology

Thirty-two in-depth telephone interviews with Quattro Pro purchasers were contacted by experienced, executive interviewers. The interviews were approximately 15 to 30 minutes in length.

An incentive of $50 was be offered to respondents to increase the cooperation rate (sample names are limited). Lotus was not identified as the sponsor of the study. Sample (a total of 90 sample pieces) for the study was obtained from the 1-2-3 Upgrade Penetration Survey and the Magellan Early Adopter Study where Quattro users have already been identified. Of the completed interviews, 14 users came from the Magellan Early Adopter list and 18 came from the 1-2-3 Upgrade list.

Because of the short turn-around time required for the survey, two suppliers were used: Gasteiger and Associates of Brighton and Keady and Associates of Boston.

Summary of Results

Best Spreadsheet

Out of the 30 users interviewed, 20 rated Quattro or Quattro Pro as the best spreadsheet on the market. Two respondents rated it tied as the best, one with 1-2-3 and one with Excel. Only 10 rated 1-2-3 the best. And, one felt SuperCalc was the best spreadsheet.

Not surprisingly, the ratings were quite different depending whether the respondent came from the 1-2-3 upgrade list or the Magellan list. Only one user of the 14 from the Magellan list rated 1-2-3 the best, whereas nine of the 18 from the 1-2-3 Upgrade list liked 1-2-3 the best.

Users had a wide variety of reasons for saying Quattro Pro the best spreadsheet:

> "It has a good preview of printing features, and I like the mouse support efficiency."
>
> ". . . because of the ability to tie-in Quattro Pro with the Paradox."
>
> ". . . with Quattro Pro, you can view more styles of graphs, more different types than you can through Lotus. And it has better color."
>
> "With Quattro it's so much easier to do graphics and report types.
>
> "It's friendlier than Lotus."
>
> "It's easy to use. The hardware requirements are not as massive as Lotus 3.1. . . . The graphics. It seems like a more user-friendly program. It's not as expensive."
>
> [4] Flexibility. Power. Compatibility. Presentation. Utilities ."

Those who preferred 1-2-3 seemed to like it better because they were accustomed to it:

> "I've been using it for years."
>
> "Because I am the most familiar with it."
>
> "It's the one I use most often."
>
> ". . . the one that has fulfilled my needs the best, has worked the best for me."
>
> "There's always been a 1-2-3 out there, so everyone know how it works and why it works."

Only a few users had specifics to explain why 1-2-3 was the best spreadsheet. In those cases, the users had just upgraded to Version 3.1.

> "Well, the new Lotus 3.1, we're pretty happy with, the one that packages Impress and everything and basically goes with a what-you-see, what-you-get type of technology and seems to have good print drivers."

Spreadsheets Being Used

Of the 32 Quattro users, 20 were still using 1-2-3 in some form or another. This again varied depending on which list they came from. Sixteen of the 18 users from the 1-2-3 Upgrade list were still using 1-2-3 versus four from the 14 users from the Magellan list.

Other spreadsheets used include: SuperCalc, ProCube, VP Planner, and Excel.

Of those who used both 1-2-3 and Quattro, the amount they used one versus the other varied widely. The numbers were split pretty evenly between those who use Quattro more, those who use 1-2-3 more, and those who use them the same amount.

When asked about the type of work they used Quattro for versus 1-2-3, the answers varied widely. The most common division of the two was using 1-2-3 for number-crunching and Quattro for the printing. However, there was one person who does exactly the opposite.

> "I use Quattro about 40% of the time and 1-2-3 60% of the time. I use 1-2-3 for day-to-day number crunching. I use Quattro for anything that needs to look nice and dressy."
>
> "I split things probably 50/50. I like 1-2-3 when I'm doing something very, very high grade, and I want to have desktop publishing. . . . I use Quattro for regular number crunching." *(Uses 1-2-3 3. 1)*

* * *

[6] In addition, many people did receive the mailing offering an upgrade from 1-2-3 to Quattro for $99. Most of them were unsure exactly how Borland got their names for the mailing; they weren't sure if Borland got it off a subscription list or really knew they used 1-2-3. It was a very compelling offer for everyone who did get it. The few who didn't act on it were those who already had Quattro.

Price

Price, particularly the $99 offer, was extremely compelling to most of the Quattro users. Sixteen of the respondents got Quattro for $99 and seven said they got it for less than $99. Of 31 respondents, only three paid more than $100 - $150, $319, $425.

When asked whether price influenced their decision to buy Quattro, 21 of the respondents said, "Yes." That was the most common reason, but not the only reason, for buying the product.

> "That offer totally influenced my decision to buy it. I probably wouldn't have bought Quattro otherwise."
>
> "It was the decisive factor in buying the Quattro, because it was worth the risk, and I know I wouldn't have bought it at full price since I already had and was using 1-2-3."
>
> "They would give you everything Lotus 2.2 can do—and more—for $99. I may be dumb, but I'm not that dumb."
>
> "There was no big incentive to switch—until they enticed me."

Hardware

Hardware requirements were a crucial decision point for a few of the users. This was less of an issue with respondents from the Upgrade list. Overall, eight mentioned Quattro's lower RAM requirements and five mentioned not having to upgrade their hardware.

Most of these users clearly could not use 1-2-3 3.0 or 3.1 on their machines. They focused on the requirements for Version 3, not Version 2.2 when comparing Quattro's hardware requirements to 1-2-3's.

Other Influencing Factors

Almost every respondent had read one or more reviews raving about Quattro Pro. Several mentioned a review in *PC Magazine* that rated it the number one spreadsheet. Several others mentioned articles in *InfoWorld*. For the most part, they did not know exactly where they had read the information, but they did know they had read good things about it.

On the other hand, very few had spoken to colleagues about the product. Most said they didn't know anyone who used the product before they had bought it. This was particularly true of those users who bought Quattro when it first came out.

The most commonly mentioned influencing reasons for buying Quattro that haven't already been mentioned were:

Graphics/graphs	9 respondents
Presentation quality	8
Compatibility with 1-2-3	6
Better interface/menus	4

Other reasons included: ease of use, on-screen help, mouse support, flexibility, colors, power.

Where Bought

Twenty-four respondents knew where they bought Quattro. The majority, 14, bought directly from Quattro. In virtually all these cases, they were taking advantage of some $99 offer for the product. Six respondents bought from a retail store, and several of them got the $99 offer there as well. It seems Egghead was also selling Quattro for $99 for a time.

Previous Spreadsheet Use

No one was using Quattro as their first-time spreadsheet. The vast majority, 24, had either used 1-2-3 before or were

still using it. Considering that half of those interviewed came from a 1-2-3 upgrade list, this was not surprising. The other previously or currently used spreadsheets were:

SuperCalc	3 respondents
VP Planner	2
ProCube	2
Excel	2

Condor 3, DB Excel, Shareware, Symphony and Visicalc—1 each

Overall Satisfaction with Quattro and 1-2-3

It is important to keep in mind that the satisfaction ratings refer to a wide variety of versions. The Quattro users were rating anything from the original Quattro to Quattro Pro 2.0. The 1-2-3 users were referring to any version from 2.01 to 3.1.

Half rated themselves "Completely satisfied" with Quattro and half rated themselves "Somewhat satisfied." Only one user rated herself "Somewhat dissatisfied." Those who rated themselves "Somewhat satisfied" either had a specific glitch or glitches they wanted fixed, or they felt nothing is perfect, so no product could make them "Completely satisfied."

Twenty-three current or former 1-2-3 users also rated their satisfaction with 1-2-3. Six were "Completely satisfied", 15 were "Somewhat satisfied", and three were "Neither satisfied nor dissatisfied." Most of those who were less satisfied were referring to version 2.01 that they used before changing to Quattro.

* * *

[11] Good reviews in magazines reinforced the respondents' desire to give Quattro a try. Most of these users were intermediate or advanced users who pay attention to what is going on in the personal computer industry. They were curious what all the fuss was about. Since such a small percentage of spreadsheet users are Quattro users, they depended on the expert opinions of the magazines instead of friends' opinions.

Once they use it, most people are very happy with Quattro.

Quattro is a good spreadsheet, and users see this. For most of them, it will do everything they need to do, in a way they want to do it. The graphics capability has a lot of appeal to many of the users. In addition, many are simply attracted to the colors and the appearance of the screen. They feel it adds a little pizzazz to the otherwise dull job of using a spreadsheet.

Hardware limitations make Quattro a necessity for some.

A small portion of the Quattro users are limited by the hardware they or their company uses. In most cases, it is an issue of RAM. All of them complain that 1-2-3 takes a tremendous amount of memory and simply can not run on their machines.

This reason may be considerably more important than this small survey showed. Since over half of those surveyed were 1-2-3 upgraders, it was likely they had enough hardware.

Users see Quattro as a 1-2-3 equivalent product for less money.

From reviews, and then later from their use, users feel that Quattro can do everything 1-2-3 can do. The most objective of them realize that Quattro can do some things better, and 1-2-3 can do other things better. However, the big difference is price. Because virtually all of those interviewed got their Quattro for less than $100, they see Quattro delivering a tremendous amount of power for a low price. While they may have seen 1-2-3's price as reasonable before, they now see an equivalent product for less money. Rationally, they feel it makes sense to go with the less expensive option.

> "With the latest version *(of 1-2-3)*, I have the impression it's a very expensive spreadsheet for very expensive hardware."

Quattro's compatibility with 1-2-3 is crucial in this 1-2-3 world.

Although many of the users have convened entirely to Quattro, its compatibility with 1-2-3 is still crucial to them. It seems very few of the users would have bought Quattro if they hadn't been able to convert their 1-2-3 files into Quattro files very easily. It continues to be important, because the users frequently exchange files with 1-2-3 users.

[12] Being able to change the menu structure to replicate the 1-2-3 menu structure is also very appealing to many users just converting from 1-2-3. Most of the Quattro users interviewed do use the menus as Quattro has set them up, but they appreciate the ability to use Quattro as 1-2-3 when they need to. One man went as far as saying that he recommends Quattro to other users, because it means they can get both Quattro and 1-2-3 in the same package.

There is still an entrenched 1-2-3 user base out there.

Several of the respondents barely used Quattro. They had bought it as a test and had decided not to switch. Most of them said they felt too familiar with 1-2-3 to try to change to something else. A few of the Quattro users had been trying actively to convert 1-2-3 users in their office to Quattro but were having no success. They said the other users were unwilling to switch. Unfortunately, it appeared that most of the 1-2-3 users did not necessarily think 1-2-3 was better, they just thought it was the product they were used to.

Recommendations

Pour all available resources into marketing Version 2.2 and the upcoming 2.3, particularly making a clear distinction between them and Version 3.1.

The product to beat Quattro Pro with is 2.3, not 3.1. Although most of the Quattro respondents were fairly advanced PC users, they were either light spreadsheet users or doing basic number crunching. Because of Lotus' focus on 3.0, and now 3.1, as the flagship product, everyone seems to

picture those products when they describe 1-2-3. In particular, these Quattro users focused on the memory and processor requirements for 1-2-3 3.0. This is *not* an issue for 2.2, but no one seems to be considering 2.2. Those with lower-end machines should feel that 1-2-3 2.2/2.3 is *the* product for their machines, not a lesser version of 3.0.

Play-up the graphics, colors, and presentation capabilities in 1-2-3 2.3.

Even if they don't use them very much, users were impressed by the "pretty" aspects of Quattro. They like the colors and the nice screens. They like being able to make impressive graphs easily. Many of them do not do intensive spreadsheets, number-crunching work, so the aspects of a spreadsheet that impress them are the easy "ooh and aah" factors. Having spreadsheet publishing capabilities in 2.3 will probably go a long way in handling the inequities.

Consider some special discounting of 2 3.

Don't give the software away for free. Offering it for free makes people suspicious that something is wrong. 1-2-3 is still seen as the de facto leader in the spreadsheet market, so Lotus shouldn't look desperate.

EXHIBIT 36

Jim Manzi

Staff Meeting
Royal Sonesta
November 16, 1990

You probably noticed that this meeting was postponed a few times. Since the last time we met was after we announced the Novell merger, we wanted to do something different this time. As you all know, this morning we announced we are acquiring Samna, the developers and marketers of the best word processing package for Windows. This is a little different from Novell.

It's smaller.

It's in the same time zone.

And most importantly this is further along. We have a final agreement rather than an intent to merge. There's more work to finalize it but we are way down the path.

We also have had our blood tests. They tested OK. And so did we.

We got our marriage license.

And we even like each others' relatives.

I'll come back to Samna in a few minutes because it's part of a large story and larger discussion.

Agenda:

1. Status of our business.
2. Challenges we face in the immediate future
3. Review our primary objectives and priorities.

I'll be followed by Frank King, Frank Moss, and Bob Schechter and then come back for questions at the end.

Current Status:

I have a few tough messages to deliver today because I think we are heading for tough economic times as you know.

There are probably 100 meetings at major companies right now holding a meeting like this one.

This meeting is going to be a lot easier than those being held by real estate developers, or mini-computer makers or major banks.

Before I continue any further, let me say right now we are not here to talk about layoffs or downsizing. We have no plans and no intentions of doing that. We are here to have a straight talk about the state of our business, our market, our competition.

And above all, we are here to talk about being smart and being prudent in this new type of environment.

Let me being with the current situation.

1. Like never before we are under attack in our core business.
   — position still strong but we have strong competitors

2. We have a big competitive exposure until next year
   — Windows vs. OS/2
   — top priority development of Windows products
   — great uncertainty around the size of this exposure

3. When there's more competition, the cost of doing business goes up
   — increased marketing and selling costs
   — increased development costs when reacting as opposed to leading

We have shipped Release 3.1 last month. What we know now:
   — sell thru data at near record levels
   — customer support volume—a good sign of early usage—already passed.
     2.2 and 3.0
   — marketing activities appear to be on track
   — live in-store upgrades
   — new advertising
   — major direct mail effort

But, the competition is stronger and smarter.

Great need to overcome our competitive challenges as quickly as possible.

First from Excel:

— While 3.1 runs under Windows, we need our full Windows product ASAP because the lack of it is costing us some business.
   — size of this exposure unknown
   — Windows is going to be extremely important: will have several windows apps next year; in fact, we will have an extremely competitive line up of apps for Windows next year.

We have a very strong competitor, as you know, in Microsoft. Let me show you how strong (SLIDE).

But this is the result of silicon steroids. Over time, I think steroids will be hazardous to your health.

Competition from Borland:

   — didn't take them seriously enough
   — targeting our upgrade business; our existing users
   — more aggressive than we were in marketing efforts

They have taken some market share points. 1 point equals $5 mm to $7 mm

Borland, or Mr. Kahn, is like the man who came to dinner. Phillipe has stayed around too long eating from our menu. (Here is a pictorial representation of what I mean). (SLIDE)

His eating shows in his waistline and his marketshare.

In Phillipe's case he does look imposing but this is a function of not too many steroids but too many donuts. And we all know that donuts are hazardous to your health long term as well.

It is time for Phillipe to leave the table.

To be a bit more serious:

Need to do whatever it takes to drive them out of our business.

This is not business as usual. These are not two ho-hum competitors. These are two very savvy companies who are trying real hard to damage us competitively and we will stop at nothing to defend and extend our spreadsheet franchise. Just as they will stop at nothing to get some of our business.

EXHIBIT 38

Houdini Analysis of Competitive Products

Automation

Definition: Most casual users see the primary benefit of a command language to be its ability to automate otherwise manual keystrokes or mouseclicks. Users who create automated macros tend to be more product-wise than technical. Macros are typically written as isolated snippets designed to speed up a tedious spreadsheet task.

Examples: Macros written at this level are often used to print ranges within a worksheet, save files, or change the style or font of a cell in a GUI spreadsheet without selecting a dialog box or menu. One of the most popular 1-2-3 macros provided an automatic switch between the numlock and enter keys as a workaround for number entry on the IBM numeric keyboard.

				Excel		
LEAF		HTalk	FImpact	Wingz	R3	
1	Developer				Casual	10

Automation

Languages Compared: Early releases of *1-2-3* were successful in encouraging casual users to automate spreadsheet routines through the straight forward mimic of first letter menu commands. *1-2-3 Release 3* offers a Learn mode that records up to 256 keystrokes and collects them into a spreadsheet cell for inclusion in a macro. The user is required to name and invoke the macro. Both *Excel, Full Impact,* and *Wingz* provide macro recorders that mimic both keystrokes and mouseclicks. They both offer automatic naming and immediate playback. *LEAF* does not offer a recorder, but it is not directed at the casual user.

HyperTalk uses different methods to provide for the non-technical, casual user. Many useful *HyperTalk* examples are shipped with HyperCard. Users can look behind any button or card in HyperCard to review its underlying script. The plain English syntax is clear and approachable. An excellent context-sensitive help system is available. Users are encouraged to give scripting a try. Wingz also uses plain English syntax in its HyperScript language, and a similar method for viewing underlying scripts.

EXHIBIT 39

To: Lotus Federal Marketing

From: Larry Roshfeld, Systems Engineer

Re: Review of Excel

Date: December 14, 1987

After recently spending some time working with Microsoft PC Excel, I thought I'd share with you my impressions of the product, based on my background as an advanced 1-2-3 user. Keep in mind that the following comments are biased, not because I work for Lotus, but rather because of how comfortable I am with 1-2-3. I will, when appropriate, attempt to point out areas of functionality in Excel that would be perceived differently by an experienced 1-2-3 user versus a novice spreadsheet user. This document is not intended as an exhaustive look at PC Excel, but rather as a way to quickly introduce you to the relative advantages and disadvantages of PC Excel compared to 1-2-3 Release 2.01 I would prefer that you not quote from this document to your customers, since the opinions expressed below are my own, and *NOT* Lotus Corporate Policy.

For those of you who would like a simple overview:

Executive Summary

PC Excel has vastly greater functionality than the current version of 1-2-3, for roughly the same price. Release 3 of 1-2-3 will add some of the features to 1-2-3 that are now currently found only in Excel, but there are some features in Excel that will not be included in 1-2-3 Release 3, and which may or may not be present in 1-2-3/G.

PC Excel requires a relatively sophisticated and expensive hardware platform. Yet even with this hardware, it "feels" slow, due to the fact that it is running under Windows 2. Paradoxically, the sophisticated 1-2-3 user who would most ben-

efit from the enhanced features in PC Excel, and who is most likely to already have the hardware required to run the product, is also the person who will have the greatest difficulty learning the product. PC Excel will be a difficult product for an experienced 1-2-3 user to learn, since its basic structure and functionality is very different from 1-2-3. It will be less difficult for a novice spreadsheet and/or PC user to learn, since they will have little to "unlearn." Existing 1-2-3 worksheet files can be brought into PC Excel, with little or no loss of formatting or functionality. PC Excel can also save files in a 1-2-3 file format.

* * *

experience, most PC users are not comfortable with a mouse. This will greatly hurt the chances of acceptance of a product, like Excel, that is very mouse dependent.

Historically, there have been a number of spreadsheet products on the market that have had greater functionality than 1-2-3. They have not made a significant dent in our market share due to a number of reasons, including the entrenchment of 1-2-3, the investment by customer organizations in training and application development, and the availability of third party templates and add-ins. While PC Excel will be able to compensate for some of the above, it probably can't overcome them. We will lose some users, and even user departments, to PC Excel, but if we do our job correctly, we should not lose any accounts.

I personally am more concerned about competition from Quattro, the new spreadsheet from Borland. It is priced at $195 retail, and offers the user a more powerful and flexible spreadsheet than 1-2-3, with the option of a "1-2-3 like" interface. I will be taking a look at Quattro in the next month, and will at that time provide you with my review of the product.

EXHIBIT 40

Memo to: Distribution List

From: Stacy Gasteiger

RE: 1-2-3 3.0 POSITIONING STUDY

The attached study summarizes the results of six focus groups we held in January to help position Release 3.0. The groups were very consistent and gave us a lot of helpful information about the upcoming product release.

There should be a half-hour edited version of the focus groups available for viewing in early March. I will let you know when you can take a look at it.

DISTRIBUTION LIST:

Liz Suneby	John Lloyd—LMS (2)
Bill Huff	Rob Perry
Laurie Phelan	Dave Barrett
Irfan Salim	Paul Pagett
Chuck Digate	Isabel Oswell—Windsor (4)
Frank Ingari	Greg Jarboe
Peter Simon	Betsey Kosheff
Scott Tucker	Tom Hughes
Linda Volpert	Carol Simpson
Jim Richard	Eileen Rudden
Kelsey Selander	Dave Pritchard
Frank Ingari	Lotus Japan

[i] 1-2-3 RELEASE 3.0 QUALITATIVE RESEARCH STUDY

EXECUTIVE SUMMARY

BACKGROUND

Lotus is currently scheduled to ship 1-2-3 Release 3.0 in mid-July, 1988, which means that all the marketing communications activities, including advertising, will begin the same

month. The initial campaign will be targeted at current 1-2-3 users, management, and DP/MIS people in Fortune 500 companies.

When 1-2-3 3.0 ships, Excel for the PC will have been on the market for nine months. Excel has been positioned as the spreadsheet for the newest technology, the 386 machine. In addition to targeting new users, it is targeting 1-2-3 users who want more power. Quattro will have been on the market for nearly a year. Borland is positioning Quattro as a full-powered spreadsheet for a significantly lower price.

Given how critical 1-2-3 is to Lotus, the product marketing group and advertising group wanted to do research to confirm or dispute their best assumptions about optimal 1-2-3 3.0 positioning in Lotus's communications, including advertising, PR, collateral, packaging, etc. Additionally, this research would give us some clues on Excel's long-term product potential.

DECISIONS TO BE MADE

To determine which of the attached positioning statements Lotus should focus on.

To determine what points about Release 3.0 best support the positioning platform.

To determine whether different positionings are required for end-users, departmental managers, or MIS/InfoCenter managers.

OBJECTIVES

To test positioning statements and support points. To determine how the positionings are rated on:

Importance
Uniqueness
Understanding
Likelihood to purchase
Overall

* * *

[iv] SUMMARY OF FINDINGS

1-2-3 Usage and Satisfaction

People appear to be generally satisfied with 1-2-3. However, with the arrival of competitive spreadsheet products, Lotus seems to be caught in a spiral of rising expectations as customers now demand more advanced capabilities.

MIS and Department managers are keenly aware of competing products and their perceived advantages, including price. What restrains those who might consider switching over is their investment in Lotus and the costs of retraining end-users.

Participants' chief perception of Excel centers around its graphic interface and design. Quattro seems to be well known primarily for its price.

On an abstract level, many people are intrigued by the new advanced features of the OS/2 and Presentation Manager technologies. However, on a more practical level, MIS and department managers are taking a cautious, wait-and-see attitude.

There is a general feeling that 3.0 represents a technological "catch up" for Lotus. At the same time, people feel that most end-users will not take advantage of the new features.

1-2-3 Upgrade History and Plans

The administrative "nightmare" of the 1A to 2.0 upgrade is one that everyone would like to forget but haven't. They say a good first step in easing the upgrade burden would be to eliminate sending back diskettes.

Everyone concedes that end-users will drive the decision to upgrade. Such a move will happen on an as needed basis and will rarely be "automatic." Assuming a simplified upgrade process, MIS will not resist upgrading.

People expect a wide range of enhancements in 3.0—faster recalculation speed, linking spreadsheets and keystroke

recording, to name a few—*but want them to work within existing memory requirements.*

Department managers, in particular, want Release 2.01 to remain available indefinitely. Because the discontinuation of IA forced many to upgrade at inconvenient times, there is concern about the fate of Release 2.01.

* * *

EXHIBIT 41

Massachusetts Mutual Life Insurance Company,

Springfield MA

To: List

Re: Quattro Evaluation

From:

Date: December 9, 1987

A preliminary evaluation of Quattro has been initiated, involving PC App Support, the Info Center, and SDD areas of Respond III and Respond Plus. Some very positive results have evolved during this testing. Quattro seems to be 100% compatible with Lotus 1-2-3 V2.x. Quattro also offers data and process integration with Paradox, MM's database standard.

If Quattro continues to prove to be an attractive alternative to Lotus, upgrades, new purchases, and custom applications may migrate to what appears to be a more powerful and versatile, and yet comparatively inexpensive solution.

Extensive testing by those areas of the company which currently utilize Lotus is needed to assist us in determining the impact of migrating to Quattro as one of our spreadsheet standards. Your specific involvement will be to test your area's current base of 123-related software.

The two key issues which should be addressed include:

Quattro's ability to run macros and templates exactly as they would with Lotus;

Quattro's compatibility with existing systems which access the spreadsheet applications (i.e. menu systems, sales illustrations, etc.)

In addition, general functionality, user interface, and an overall assessment of the product as it relates to your clients' needs should also be considered.

We will contact you by Friday, December 11, 1987 to make arrangements for you to acquire a production copy of Quattro. We would like the evaluation process to be completed by year-end. A meeting will be scheduled in early January to exchange ideas and address issues.

If anyone needs special assistance or further information, please do not hesitate to call me (x3190), Diane Giuliana (x5239) or Bridget Knudsen (x2451).

Thanks

List:	Nancy Crosby	L398
	Susan Gay	A255
	Jim Snyder	L403
	Denise Szczebak	D301
	Elaine Carlson	D296
	Annora Stewart	MMH3
FYI:	Candice Kasai	B060
	Diane Giuliana	B303
	Bridget Knudsen	B303

(Handwritten material illegible)

Exhibit in Support of Borland International, Inc.'s Motion for Reconsideration, dated August 31, 1992 (Declaration of Andrew Konstantaras (Dkt. No. 200))

EXHIBIT I

F9: CALC	Recalculate worksheet
F10: GRAPH	Draw graph according to most recent graphing specifications

"1-2-3 COMMANDS"

"W"orksheet Commands

"G"lobal	Controls over-all settings
"F"ormat	sets the display appearance of numbers
"F"ixed	fixed # of decimal pts., 0-15
"S"cientific	exponential (1.2E+01)
"C"urrency	$ and commas ($1,000)
"," (comma)	(1,000.546)
"G"eneral	drops trailing 0 after decimal pt.
"t"/- (graph)	displays ++++ or ———
"P"ercent	divides by 100, shows % (10%)
"D"ate	controls display of date numbers
"1"	(20-Jun-48)
"2"	(20-Jun)
"3"	(Jun-48)
"T"ext	shows formulas as "text" (+B2tC3)
"L"abel-Prefix	controls label alignment in cell
"L"eft	label is preceded by ' (a preceding
"R"ight	label is preceded by " \ creates re-
"C"enter	label is preceded by ^ peating label)
"C"olumn-width	may be 1-72 characters wide
"R"ecalc	controls recalculation options
"N"atural	recalcs formulas in dependency order
"C"olumnwise	recalcs left to right, down columns
"R"owwise	recalcs top to bottom, across rows
"A"utomatic	recalcs every time new entry is made
"M"anual	only recalcs when F9 [CALC] key is pressed
"I"teration	sets # of times recalc occurs
"P"rotection	controls access to protected cells
"E"nable	turns protection on

"D"isable	turns protection off
"D"efault	controls equipment settings
"P"rinter	controls printer options
"I"nterface	choose Serial or Parallel cards
"A"uto-LF	Line Feed after Carriage Return?
"L"eft	margin spaces from edge of page
"R"ight	total # spaces from left edge
"T"op	margin from top of paper
"B"ottom	margin from bottom of paper
"P"age-Length	total # lines on paper
"W"ait	pause after each page?
"S"etup	send special control string
"Q"uit	return to "D"efault submenu
"D"isk	sets initially current disk
"N"one	program disk's drive is current
"A B C D E F G"	specified drive is current
"S"tatus	shows current "D"efault settings
"U"pdate	records changes in "D"efault submenu
"Q"uit	return to "G"lobal submenu
"I"nsert	adds blank, unformatted cells
"C"olumn	
"R"ow	
"D"elete	removes cell contents and formats
"C"olumn	deletes entire column
"R"ow	deletes entire row
"C"olumn-width	sets width of pointer's column
"S"et	sets new width
"R"eset	returns column to "G"lobal setting
"E"rase	erases entire worksheet
"T"itles	keeps title cells visible on screen
"H"orizontal	keeps specified rows visible
"V"ertical	keeps specified columns visible
"B"oth	keeps both rows and columns visible
"C"lear	eliminates all titles
"W"indow	splits the display window in two
"H"orizontal	splits window at pointer's row
"V"ertical	splits window at pointer's column
"S"ync	makes windows scroll together

"U"nsync	lets windows scroll independently
"C"lear	returns to single window display
"S"tatus	shows current "G"lobal settings

"/R"ange Commands

"F"ormat	sets display appearance of values in range
"F"ixed	fixed # of decimal pts., 0-15
"S"cientific	exponential (1.2E+01)
"C"urrency	$ and commas ($1,000)
"," (comma)	(1,000.546)
"G"eneral	drops trailing 0 after decimal pt.
"t"/- (graph)	displays ++++ or ————
"P"ercent	divides by 100, show % (10%)
"D"ate	controls display of date numbers
"1"	(20-Jun-48)
"2"	(20-Jun)
"3"	(Jun-48)
"T"ext	shows formula as "text" (+B2tC3)
"R"eset	returns range to "G"lobal format
"L"abel-Prefix	sets alignment of labels already in range
"L"eft	(')
"R"ight	(")
"C"enter	(^)
"E"rase	erases contents of specified range
"N"ame	gives range addresses a name
"C"reate	assigns name to use instead of addresses
"D"elete	un-names a range
"L"abels	makes label the name of adjacent cell
"L"eft "R"ight "U"p "D"own	
"R"eset	erases all range names
"J"ustify	changes width of text paragraph
"U"nprotect	removes or replaces protection facility
"P"rotect	around cells in range
"I"nput	pointer can only go to unprotected cells

"/C"opy Command
copies range contents to new location(s)

"/M"ove Command
moves range contents to new location

"/F"ile Commands
"S"ave — sends current worksheet to disk file
"R"etrieve — brings worksheet file from disk to screen
"C"ombine — overlays disk file on current worksheet
"C"opy) "E"ntire-file — incoming cells replace worksheet's
"A"dd) "N"amed-Range — overlapping cells contents summed
"S"ubtract) — incoming values minus worksheet's
"X"tract — sends part of current worksheet to disk file
"F"ormula — saves formulas in Xtract
"V"alues — only saves current value of formulas
"E"rase — erases one or more disk files
"W"orksheet — erases worksheet (.WK5) file(s)
"P"rint — erases print (.PPN) file(s)
"G"raph — erases graph/picture (.PIC) file(s)
"L"ist — lists names of files, shows free disk memory
"W"orksheet — lists worksheet (.WK5) files on current disk
"P"rint — lists all print (.PPN) files on disk
"G"raph — lists graph/picture (.PIC) files on disk
"I"mport — brings print/text (.PPN) disk file to screen
"T"ext — only brings text from disk to screen
"N"umbers — brings numbers and "quoted" text to screen

"D"isk — switches current disk drive

"/P"rint Commands
("P"rinter) — sends worksheet to printer
("F"ile) — "prints" worksheet in print (.PPN) file
"R"ange — sets the range to be printed
"L"ine — advances printer one blank line
"P"age — advances printer to top of next page
"O"ptions — controls appearance of printed page
"H"eader — inserts extra characters at top and/or bottom of printed page
"F"ooter
"M"argins — controls blank space at edges of paper
"L"eft "R"ight "T"op "B"ottom
"B"orders — prints specified columns/rows along left side and/or at top of each page of range being printed
"C"olumns
"R"ows
"S"etup — sends special control characters to printer
"P"age-Length — tells 1-2-3 total # of lines on page
"O"ther
"A"s-Displayed — printout copies screen display
"C"ell-Formulas — printout shows formula text, not values
"U"nformatted — suppresses/reactivates the use of headers, footers & page breaks
"F"ormatted
"Q"uit — return to main print menu
"C"lear — eliminates/resets print settings to "/WGD"
"R"ange — "forgets" last specified range
"B"orders — cancels repeating borders
"F"ormat — resets page-length; margins, set-up string
"A"ll — resets all of above to global default values
"A"lign — signifies that paper has been repositioned
"G"o — prints the selected range with options
"Q"uit — return to Ready Mode

"/G"raph Commands

"T"ype	selects type of graph displayed
"L"ine	graphs 1 to 6 series of numbers
"B"ar	graphs 1 to 6 series of numbers
"X"Y	graphs relationship of 2 series of numbers
"S"tacked-Bar	graphs numbers in proportion to their sum
"P"ie	graphs one series of numbers in prop. to sum
"X A B C D E G"	defines label and data ranges for graphs
"R"eset	returns settings to default values
"G"raph	erases current graph's specifications
"X A B C D E F G"	removes range contents from current graph
"Q"uit	return to main "/G"raph menu
"V"iew	display current graph on screen
"S"ave	saves current graph specs in disk file
"O"ptions	
"L"egend	labels each data range
"A B C D E F"	
"F"ormat	sets method of displaying/connecting
"G"raph)	data points for entire graph or
"A") "L"lines	individual data ranges
"B") "S"ymbols	
"C") "B"oth	
"D") "N"either	
"E"	
"F"	
"Q"uit	return to main "/G"raph menu
"T"itles	sets title for axes and/or graph
"F"irst	when printed, 1st graph title is double-sized
"S"econd	second line of graph title
"X"-axis	titles horizontal axis
"Y"-axis	titles vertical axis
"G"rid	displays scale lines across graph

"H"orizontal	displays horizontal lines at tick marks
"V"ertical	displays vertical lines at tick marks
"B"oth	displays both horizontal and vertical lines
"C"lear	erases all grid lines
"S"cale	sets display scale of axes
("X"-axis)	only relevant for XY graphs
("Y"-axis)	not used with pie graphs
"A"utomatic	includes min. and max. data values
"M"anual	focuses in on particular area of concern
"L"ower	with manual scaling, sets lower scale value
"U"pper	with manual scaling, sets upper scale value
"F"ormat	sets format of scaling numbers
"F"ixed	fixed # decimal pts., 0-15
"S"cientific	exponential (1.2E+01)
"C"urrency	$ and commas ($1,000)
"," (comma)	(1,000.546)
"G"eneral	drops trailing 0 after decimal pt.
"t"/- (graph)	displays ++++ or ————
"P"ercent	divides by 100, shows % (10%)
"D"ate	controls display of date numbers
"1"	(20-Jun-48)
"2"	(20-Jun)
"3"	(Jun-48)
"T"ext	shows formula as "text" (+B2tC3)
"Q"uit	return to options submenu
"C"olor	displays graph bars, lines, symbols in color
"B"&W	displays graph bars, lines, symbols in b&w
"D"ata-labels	sets cells to label graphed data-points
"A B C D E F"	specifies data range to be labeled
"Q"uit	return to options submenu
"Q"uit	return to main graph menu

"N"ame	names graph specs to be saved/ retrieved
"C"reate	names and saves current specifications
"U"se	retrieves named graph, even if data is changed
"D"elete	removes name (forgets) specified graph specs
"R"eset	eliminates all graph spec. names
"Q"uit	return to Ready Mode

"/D"ata Commands

"F"ill	fills range with incremented numbers
"T"able	shows how dif. entries affect formula values
"1"	puts series of values in series of formulas
"2"	puts 2 series of values in one formula
"S"ort	sorts a list or database of entries
"D"ata-Range	range whose rows (records) will be sorted
"P"rimary-Key	the column (field) used as 1st basis of sort
"S"econdary-Key	the column (field) used a 2ndary basis of sort
"R"eset	cancels specified data-range and sort keys
"G"o	triggers the sort
"Q"uit	return to main "/D"ata menu
"Q"uery	performs operations on a database
"I"nput	specifies the input range
"C"riterion	specifies the criterion range
"O"utput	specifies the output range
"F"ind	finds data that meets criterion
"E"xtract	copies criteria-specified data to output range
"D"elete	erases criteria-specified data from database

"U"nique	copies one copy of each record in output range
"R"eset	cancels input, criterion & output range specs.
"Q"uit	return to Ready Mode
"D"istribution	calculates frequency distribution of values

"/Q"uit Command

leave 1-2-3, return to Access or DOS

GRAPH PROGRAM COMMANDS

"S"elect	selects graphs for printing
"O"ptions	selects formatting options for graphs
"C"olor	assigns colors to parts of graphs
"G"rid	assigns color to gridwork and text
"A"	assigns color to data range A
"B"	assigns color to data range B
"C"	assigns color to data range C
"D"	assigns color to data range D
"E"	assigns color to data range E
"F"	assigns color to data range F
"Q"uit	exits Color submenu
"B"lack	
"R"ed	
"O"range	
"Y"ellow	
"G"reen	possible colors for each data range
"B"lue	
"P"urple	
"B"rown	
"F"ont	selects fonts for graphs
"1"	selects font for 1st line of text (will be used for rest of graph if no font 2 is specified)
"2"	selects font for rest graph
"S"ize	controls size and placement of graph
"F"ull	produces full-page graph
"H"alf	produces half-page graph

"M"anual	adjusts size settings manually
"L"eft	controls left margin
"T"op	controls top margin
"W"idth	controls width of graph
"H"eight	controls height of graph
"R"otation	controls rotation of graph
"Q"uit	exits Size submenu
"P"ause	controls pause between printing graphs
"E"ject	controls page eject between printing graphs
"Q"uit	exits Options menu
"G"o	prints selected graphs
"C"onfigure	configures GRAPH to system hardware
"F"iles	selects drives to search for files
"P"ictures	selects drive to search for Picture files
"F"onts	selects drive to serach for Font files
"Q"uit	exits Files submenu
"D"evice	selects graphics device for printing
"1"	selects Epson MX-80/100 (single) as device
"2"	selects Epson MX-80/100 (double) as device
"3"	selects IDS Prism printer as device
"4"	selects HP 7470A plotter as device
"P"age	selects default page size
"L"ength	selects default page length
"W"idth	selects default page width
"Q"uit	exits Page submenu
"I"nterface	selects interface for graphics device
"P1"	First Parallel interface (LPT1:)
"S1"	First Serial interface (COM1:)
"P2"	Second Parallel interface (LPT2:)
"S2"	Second Serial interface (COM2:)
"S"ave	record configuration on disk
"C"ancel	cancel Save

"R"eplace	replace old configuration with new
"R"eset	resets "unsaved" configuration settings to default
"Q"uit	exits Configure submenu
"A"lign	make current paper position "top of form"
"P"age	advance paper to top of next page
"Q"uit	exit GRAPH program
"Y"es	exit from GRAPH to DOS or LOTUS ACCESS SYSTEM
"N"o	return to GRAPH program

"@ FUNCTIONS"

"Mathematical Functions"

@ABS(x)	absolute value
@ACOS(x)	arc cosine
@ASIN(x)	arc sine
@ATAN(x)	2-quadrant arc tangent
@ATAN2(x, y)	4-quadrant arc tangent
@COS(x)	cosine
@EXP(x)	exponential
@INT(x)	integer part

Documentary Exhibits submitted at Trial,
filed May 5, 1993
(Dkt. No. 359)

J.A. 899

EXHIBIT 7

QUATTRO PRO 3.0 USER'S GUIDE

SUPERIOR SPREADSHEET POWER

BORLAND

APPENDIX

Commands in Lotus 1-2-3 and the Quattro Pro equivalents

The table in this appendix lists the commands in Lotus 1-2-3, identifying the corresponding commands in Quattro Pro. You can use it to adjust to the Quattro Pro menu tree. For example, if you know where a command is in 1-2-3, but you don't know the equivalent command in Quattro Pro, you can look it up here.

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents

Command in 1-2-3 *v.* 2.01	Quattro Pro command
WORKSHEET	
Global	
Format	/Options | Formats | Numeric Format
Label Prefix	/Options | Formats | Align Labels
Column Width	/Options | Formats | Global Width
Recalculation	/Options | Recalculation
Natural	/O | R | Order | Natural
Columnwise	/O | R | Order | Columnwise
Rowwise	/O | R | Order | Rowwise
Automatic	/O | R | Mode | Automatic
Manual	/O | R | Mode | Manual
Background	/O | R | Background
Iteration	/O | R | Iteration
Protection	/Options | Protection
Default	
Printer	
Interface	/Options | Hardware | Printers | 1st (or 2nd) Printer | Device
Auto LF	/Options | Hardware | Printers | Auto LF
Left	/Print | Layout | Margins | Left
Right	/Print | Layout | Margins | Right
Top	/Print | Layout | Margins | Top
Bottom	/Print | Layout | Margins | Bottom
Page Length	/Print | Layout | Margins | Page Length
Wait	/Options | Hardware | Printers | Single Sheet
Setup	/Print | Layout | Setup String
Name	/Options | Hardware | Printers | Default Printer
Directory	/Options | Startup | Directory
Status	
Update	/Options | Update
Other	
International	/Options | International
Punctuation	/Options | International | Punctuation
Currency	/Options | International | Currency
Date	/Options | International | Date
Time	/Options | International | Time
Help	
Clock	/Options | Other | Clock
Zero	/Options | Formats | Hide Zeros

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 *v.* 2.01	Quattro Pro command
WORKSHEET	
Insert	
Column	/Edit | Insert | Columns
Row	/Edit | Insert | Rows
Delete	
Column	/Edit | Delete | Columns
Row	/Edit | Delete | Rows
Column	
Set Width	/Style | Column Width
Reset Width	/Style | Reset Width
Hide	/Style | Hide Column | Hide
Display	/Style | Hide Column | Expose
Erase	/File | Erase
Titles	/Window | Options | Locked Titles
Window	/Window | Options
Horizontal	/Window | Options | Horizontal
Vertical	/Window | Options | Vertical
Sync	/Window | Options | Sync
Unsync	/Window | Options | Unsync
Clear	/Window | Options | Clear
Status *(display only—commands listed will change these settings)*	
Normal Memory	/Options | Hardware
Expended Memory	/Options | Hardware
Co-Processor	/Options | Hardware
Recalc Mode	/Options | Recalculation | Mode
Recalc Order	/Options | Recalculation | Order
Iterations	/Options | Recalculation | Iteration
Circular Reference	/Options | Recalculation | Circular Cell
Cell Format	/Options | Formats | Numeric Format
Label Prefix	/Options | Formats | Align Labels
Column Width	/Options | Formats | Global Width
Suppress Zero	/Options | Formats | Hide Zeros
Protection	/Options | Protection
Macro Library	/Tools | Macro | Library
Page	/Style | Insert Break

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 *v.* 2.01	Quattro Pro command
RANGE	
Format	/Style	Numeric Format
Label	/Style	Alignment
Left	/Style	Alignment	Left
Right	/Style	Alignment	Right
Center	/Style	Alignment	Center
Erase	/Edit	Erase Block
Name	/Edit	Names
Create	/Edit	Names	Create
Delete	/Edit	Names	Delete
Labels	/Edit	Names	Labels
Reset	/Edit	Names	Reset
Table	/Edit	Names	Make Table
Justify	/Tools	Reformat
Protect	/Style	Protection	Enable
Unprotect	/Style	Protection	Disable
Input	/Database	Restrict Input
Value	/Edit	Values
Transpose	/Edit	Transpose
COPY	/Edit	Copy
MOVE	/Edit	Move
FILE	
Retrieve	/File	Retrieve
Save	/File	Save
Combine	/Tools	Combine
Copy	/Tools	Combine	Copy
Entire File	/Tools	Combine	Copy	File
Named Range	/Tools	Combine	Copy	Block
Add	/Tools	Combine	Add
Entire File	/Tools	Combine	Add	File
Named Range	/Tools	Combine	Add	Block
Subtract	/Tools	Combine	Subtract
Entire File	/Tools	Combine	Subtract	File
Named Range	/Tools	Combine	Subtract	Block
Xtract	/Tools	Xtract
Erase	/Edit	Erase (in a File Manager window)
Worksheet	
Print	
Graph	
Other	

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 *v.* 2.01	Quattro Pro command
FILE	
List	/Sort (in a File Manager window)
Worksheet	
Print	
Graph	
Other	
Import	/Tools	Import
Text	/Tools	Import	ASCII Text File
Numbers	/Tools	Import	Comma & "" Delimited File
Comma	/Tools	Import	Comma & "" Delimited File
Directory	/File	Directory
PRINT	
Printer	/Print	Destination	Printer
Range	/Print	Block
Line	/Print	Adjust Printer	Skip Line
Page	/Print	Adjust Printer	Form Feed
Options	/Print	Layout
Header	/Print	Layout	Header
Footer	/Print	Layout	Footer
Margins	/Print	Layout	Margins
Left	/Print	Layout	Margins	Left
Right	/Print	Layout	Margins	Right
Top	/Print	Layout	Margins	Top
Bottom	/Print	Layout	Margins	Bottom
Borders	/Print	Headings
Columns	/Print	Headings	Top Heading
Rows	/Print	Headings	Left Heading
Setup	/Print	Layout	Setup String
Pg Length	/Print	Layout	Margins	Page Length
Other	
As Displayed	/Print	Format	As Displayed
Cell Formulas	/Print	Format	Cell Formulas
Formatted	/Print	Layout	Break Pages	Yes
Unformatted	/Print	Layout	Break Pages	No
Update	/Print	Layout	Update
Clear	/Print	Layout	Reset
All	/Print	Layout	Reset	All
Range	/Print	Layout	Reset	Print Block
Borders	/Print	Layout	Reset	Borders
Format	/Print	Layout	Reset	Layout
Align	/Print	Adjust Printer	Align
Go	/Print	Spreadsheet Print
File	/Print	Destination	File
(same options as /Print | Printer)	

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 v. 2.01	Quattro Pro command
GRAPH	
Type	/Graph | Graph Type
X	/Graph | Series | X-Axis Series
A – F	/Graph | Series | 1st – 6th Series
Reset	
Graph	/Graph | Customize Series | Reset | Graph
X	/G | C | Reset | X-Axis
A – F	/G | C | Reset | 1st – 6th Series
View	/Graph | View
Save	/Print | Graph Print | Write Graph File
Options	
Legend	/Graph | Text | Legends
A – F	/Graph | Text | Legends | 1st – 6th Series
Format	/G | C | Markers & Lines | Formats
Graph	/G | C | M | Formats | Graph
A – F	/G | C | M | Formats | 1st – 6th Series
Markers	/G | C | Markers & Lines | Markers
A-F	/G | C | M | Markers | 1st – 6th Series
Titles	/Graph | Text
First	/Graph | Text | 1st Line
Second	/Graph | Text | 2nd Line
X-Axis	/Graph | Text | X-Title
Y-Axis	/Graph | Text | Y-Title
Grid	/Graph | Overall | Grid
Scale	
Y-Scale	/Graph | Y-Axis
Automatic	/Graph | Y-Axis | Scale | Automatic
Manual	/Graph | Y-Axis | Scale | Manual
Lower	/Graph | Y-Axis | Low
Upper	/Graph | Y-Axis | High
Format	/Graph | Y-Axis | Format of Ticks
Indicator	/Graph | Y-Axis | Display Scaling
X-Scale	/Graph | X-Axis
Automatic	/Graph | X-Axis | Scale | Automatic
Manual	/Graph | X-Axis | Scale | Manual
Lower	/Graph | X-Axis | Low
Upper	/Graph | X-Axis | High
Format	/Graph | X-Axis | Format of Ticks
Indicator	/Graph | X-Axis | Display Scaling
Scale	
Skip	
Color	/Graph | Overall | Color/B&W | Color
B&W	/Graph | Overall | Color/B&W | B&W
Data Labels	/Graph | Customize Series | Interior Labels
A – F	/G | C | Interior Labels | 1st – 6th Series
Update	/Graph | Customize Series | Update

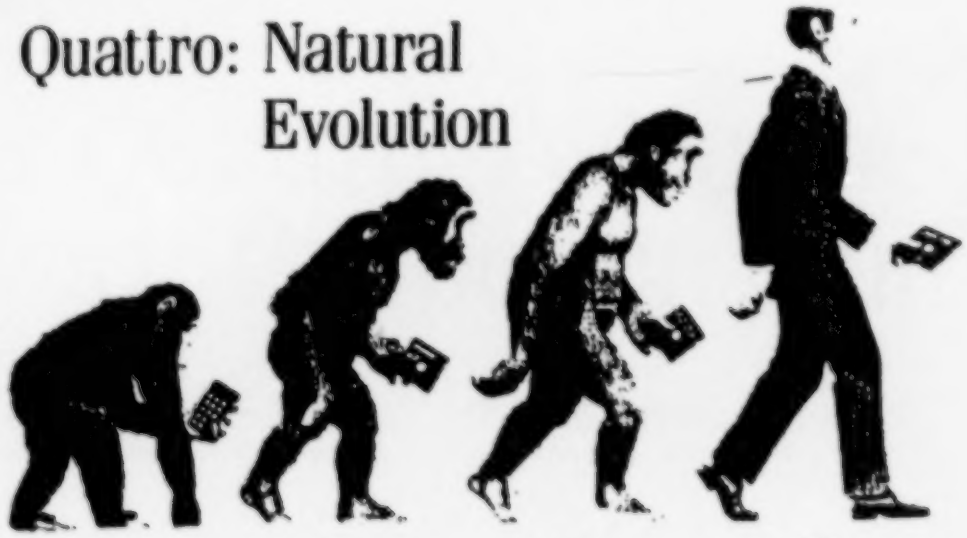
Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 v. 2.01	Quattro Pro command
GRAPH	
Name	/Graph | Name
Use	/Graph | Name | Display
Create	/Graph | Name | Create
Delete	/Graph | Name | Erase
Slide	/Graph | Name | Slide
Reset	/Graph | Name | Reset
DATA	
Fill	/Edit | Fill
Table	/Tools | What-If
1	/Tools | What-If | 1 Variable
2	/Tools | What-If | 2 Variables
Reset	/Tools | What-If | Reset
Sort	/Database | Sort
Data Range	/Database | Sort | Block
Primary Key	/Database | Sort | 1st Key
Secondary Key	/Database | Sort | 2nd Key
Reset	/Database | Sort | Reset
Go	/Database | Sort | Go
Query	/Database | Query
Input	/Database | Query | Block
Criterion	/Database | Query | Criteria Table
Output	/Database | Query | Output Block
Assign Names	/Database | Query | Assign Names
Find	/Database | Query | Locate
Extract	/Database | Query | Extract
Unique	/Database | Query | Unique
Delete	/Database | Query | Delete
Reset	/Database | Query | Reset
Distribution	/Tools | Frequency
Matrix	
Invert	/Tools | Advanced Math | Invert
Multiply	/Tools | Advanced Math | Multiply
Regression	/Tools | Advanced Math | Regression
X-Range	/T | A | Regression | Independent
Y-Range	/T | A | Regression | Dependent
Output Range	/T | A | Regression | Output
Intercept	/T | A | Regression | Y Intercept
Reset	/T | A | Regression | Reset
Go	/T | A | Regression | Go

Table I.1: Commands in 1-2-3 and their Quattro Pro equivalents (continued)

Command in 1-2-3 *v.* 2.01	Quattro Pro command
DATA	
Parse	/Tools | Parse
Format Line	
Create	/Tools | Parse | Create
Edit	/Tools | Parse | Edit
Input Column	/Tools | Parse | Input
Output Range	/Tools | Parse | Output
Reset	/Tools | Parse | Reset
Go	/Tools | Parse | Go
SYSTEM	
OS	/File | Utilities | DOS Shell
QUIT	/File | Exit

Quattro: Natural Evolution



1978 HAND-HELD CALCULATOR — 1979 VISICALC® — 1982 LOTUS 1-2-3® — 1987 QUATTRO®

Quattro: The Professional Spreadsheet

	FEATURE	QUATTRO	LOTUS 2.01
SPEED	ReCalc Cash Flow Model (5K cells)	.27 sec.	2.90 sec.
	Delete Row (5K cells (Recalc Time)	.76 sec.	2.38 sec.
	Load File (15K cells)	16.8 sec.	19.8 sec.
	Page Down (A1 to A1000)	12.2 sec.	17.4 sec.
GRAPHICS	Presentation-quality Graphics	YES	NO
	Graph Types	18	6
	Integrated Graph Printing	YES	NO
	Full Graph Customization	YES	NO
	On-Screen Font Styles	11	1
	PostScript Support	YES	NO
VERSATILITY	User-modifiable Menus	YES	NO
	Menu Shortcuts	YES	NO
	Pull-down menus	YES	NO
	Point and Press Editing	YES	NO
	Automatic installation	YES	NO
POWER	Macro Learn Mode	YES	NO
	Maximum Number of Macros	Unlimited	27
	Single Step Macro Debugging Environment	YES	NO
	Price	$199.95	$495

60-Day Money-Back Guarantee*

Benchmark details available upon request.

Get Quattro, the professional spreadsheet for only $199.95

Quattro is so advanced it's easy to use and it's less than half the price of 1-2-3. It's fully compatible with all your existing 1-2-3 files and macros—but it makes everything in them look better, print better and makes your work go faster.

For the dealer nearest you, a brochure, or to order now,
Call (800) 543-7543

BORLAND INTERNATIONAL

For the IBM PS/2 and the IBM family of personal computers and all 100% compatibles

INTRO

QUATTRO

CONFIDENTIAL
503897

Borland's New Professional Spreadsheet

Quattro: Twice the speed. Twice the power. Half the price.

Quattro™, our new professional spreadsheet, proves there are better and faster ways to do everything. To do graphics. To recalculate. To do macros. To save and retrieve. To search, sort, load. To do anything and everything that state-of-the-art spreadsheets should do.

Quattro gives you presentation-quality graphics

Quattro brings new highs in quality graphics to your spreadsheet. It also brings Postscript™ support and new variety and diversity to the kinds of graphs and graphics you can produce from your spreadsheet, and you can produce hard copy of your graphics—with either printer or plotter—without leaving the spreadsheet. All you do is hit "Print." Quattro makes it easy to get hard copy—and you don't have to buy a separate graphics program.

Quattro recalculates much faster than you-know-who

The smartest and fastest way to recalculate a spreadsheet is to do what Quattro does, something called "intelligent recalc," which in English means you only re-count the numbers that count.

In a spreadsheet, not all numbers are born equal and changing one number doesn't always change *everything*, so Quattro recalculates just the formulas that matter, not all the formulas it knows. (You wouldn't reshoot a whole movie just because you changed one scene, but unfortunately, that's the way 1-2-3 does it—and that's why it takes so long.)

Quattro demystifies Macros and makes your work go faster

Using macros—electronic shortcuts—is easy with Quattro. Quattro offers a complete macro debugging environment as you "single-step" through your macros and record them as you work.

If you know how to use 1-2-3, you know how to use Quattro

You don't have to learn a whole new program. Quattro works directly with all 1-2-3 file formats. No importing/exporting or macro translation is required. Quattro can also load and save ASCII, Paradox®, and dBASE® files.
Compatible with 1-2-3? Yes.
Faster than 1-2-3? Yes.
Technically superior to 1-2-3? Yes.
Half the price of 1-2-3? Yes!

Quattro includes SQZ!® Plus data compression

A special implementation of SQZ! Plus, the spreadsheet file compression utility, is built into Quattro and comes to you absolutely free! SQZ! Plus for Quattro automatically compacts and expands Quattro spreadsheets by up to 95% during file saving and retrieving.

CONFIDENTIAL

503898

J.A. 908

PLAINTIFF'S EXHIBIT 18

[WESTERN UNION]

BORLAND INTERNATIONAL
P. O. BOX 1866
SANTA CRUZ, CA 95061-1866

004899

HENRY GUTMAN
OSULLIVAN GRAEV KARABEL
30 ROCKEFELER PLZ
NEW YORK, NY 10112

RECENTLY WE OFFERED YOU THE CHANCE TO UPGRADE* TO QUATTRO® PRO—BORLAND'S EXTRAORDINARY NEXT-GENERATION SPREADSHEET— FOR THE LIMITED-TIME PRICE OF $99.95. IN CASE YOU MISSED OUR MAILING, WE'D NOW LIKE TO GIVE YOU ONE MORE CHANCE TO ORDER THIS $495 PROGRAM AT OUR SPECIAL PRICE.

WE HAVE EXTENDED OUR DEADLINE TO AUGUST 30, 1990. UNTIL THEN, YOU CAN STILL PAY JUST $99.95 FOR A SPREADSHEET THAT OFFERS YOU:

- EXCEPTIONAL COMPATIBILITY WITH LOTUS® 1-2-3®. USE THE SAME COMMANDS AND MACROS AS YOU WOULD IN 1-2-3. QUATTRO PRO WILL ALSO READ, WRITE AND LOAD EVEN THE LARGEST FILES CREATED WITH 1-2-3 RELEASES 1A AND 2.01.
- MULTI-PAGE SPREADSHEET CONSOLIDATION. USE "HOT LINKS" TO COMBINE INFORMATION FROM UP TO 64 LOADED AND UNLOADED SPREADSHEETS WITHOUT CUTTING AND PASTING.
- ADVANCED SPREADSHEET PUBLISHING. QUATTRO PRO CONTAINS PUBLISHING FEATURES RIVALING

THOSE OF EXPENSIVE, STAND-ALONE GRAPHICS PACKAGES.

BEST OF ALL, QUATTRO PRO COMES WITH A 60-DAY MONEY-BACK GUARANTEE.

IF YOU HAVE ALREADY RESPONDED TO OUR INITIAL OFFER, PLEASE EXCUSE US FOR CONTACTING YOU AGAIN. BUT IF YOU HAVE NOT, REMEMBER, YOU ONLY HAVE UNTIL AUGUST 30, 1990 TO UPGRADE TO THIS $495 PROGRAM FOR JUST $99.95. I URGE YOU TO ORDER QUATTRO PRO TODAY.

PHILIPPE KAHN
PRESIDENT, BORLAND INTERNATIONAL

TO ORDER: RETURN THIS FORM IN THE ENCLOSED ENVELOPE. FOR FASTER SERVICE CALL US TOLL-FREE AT 1 (800) 331-0877. OR VISIT ANY PARTICIPATING SOFTWARE DEALER.*

HENRY GUTMAN ELIGIBILITY # 59124569 QP97
OSULLIVAN GRAEV KARABEL
30 ROCKEFELER PLZ
NEW YORK, NY 10112

IMPORTANT! PLEASE INDICATE DISK SIZE: ( ) 3 1/2" ( ) 5 1/4"

QUATTRO PRO	$99.95
STATE SALES TAX**	_____
SHIPPING	5.00
TOTAL	$_____

( ) BILL ME ( ) CHECK ENCLOSED
( ) MASTERCARD ( ) VISA ( ) AMEX
CARD # _____-_____-_____-_____
EXPIRATION DATE _____-_____

*Eligibility limited to current orders of 1-2-3, Symphony®, Excel®, or SuperCalc®. You will be required to show proof of ownership after receiving Quattro Pro. To qualify at retail, present an original

manual page or disk from any version of 1-2-3, Symphony, SuperCalc or Excel.

** Applicable in: CA, CT, GA, IL, MA, MI, NY, OH, PA, TX, VA, WA.

Purchase orders: Attach this certificate to a preprinted P. O. form.
Terms: Net 30 days. Point of shipment: Scotts Valley, CA. Subject to approval.

PLAINTIFF'S EXHIBIT 19

BORLAND SCHOLAR
PROGRAM-INSTRUCTOR COUPON

* * *

BORLAND SCHOLAR PROGRAM
1800 Green Hills Road, P. O. Box 660001
Scotts Valley, CA 95067-0001

September 6, 1990

Instructor Certificate #:
60214167

Ms. Dorothy Konrad
English
Barat College
Lake Forest, IL 60045

Dear Ms. Konrad:

Several months ago we introduced QUATTRO®PRO, the next-generation spreadsheet that's being hailed as the best ever by educators, the computer industry, and major corporations.

Because you are a valued member of the educational community, Borland is making you a very special offer:

> For a limited time, you can order the complete $495 version of QUATTRO PRO for only $69.95. That's more than 85% off the regular retail price.
>
> Best of all, this offer is 100% risk-free. Your QUATTRO PRO includes our standard 60-day, money-back guarantee.
>
> And, because we want to extend this extraordinary value to your students, we've enclosed four Student Discount Certificates, good for the same special deal. (Additional Student Coupons are available on request).

Most likely, you and your students are currently using Lotus® 1-2-3®. And if you're reluctant to change, I understand.

That's why we've made QUATTRO PRO fully compatible with 1-2-3 version 2.01 file formats. You can adapt your copy of QUATTRO PRO to be compatible with the commands and macros familiar to 1-2-3 users, and best of all, run all your existing courseware with virtually no modification.

Furthermore, QUATTRO PRO offers you and your students the most important advanced features of 1-2-3 (versions 2.2 and 3.0), Excel®, and Supercalc® all in a single program without requiring you to purchase expensive new PC hardware. These features include:

- Advance spreadsheet publishing. Turn your numbers into exciting charts and graphs, which can be electronically linked and pasted into your spreadsheet. Embellish your work with publishing features built-in drawing tools; color, shading, and patterns; text in different sizes and styles (including nine Bitstream™ fonts), and the PicturePak™ clip-art library that rival those of expensive stand-alone graphics packages. Then link your graphs and charts into a single integrated slide presentation.
- Powerful analytical tools. QUATTRO PRO is the first spreadsheet that, by itself, meets the rigorous requirements of the business academic community, offering linear programming, regression analysis, matrix mathematics, and X/Y logarithmic graphing.
- Multi-page spreadsheet consolidation. Take any cells in the spreadsheet you're using and create "hot links" to cells in up to 64 other spreadsheets whether loaded in RAM or on disk. This means you can reference any cell from any spreadsheet from any formula you're working in. When you make a change in one cell, it's immediately reflected in all the other linked cells.

How can QUATTRO PRO do all this on an ordinary PC? The secret lies in Borland's new patent-pending VROOMM technology. VROOMM (Virtual Runtime Object-Oriented Memory Manager) intelligently directs QUATTRO PRO to load only the program components you're likely to need as you work. This leaves you enough memory to build and handle large, complex spreadsheets without sacrificing excellent performance.

We at Borland believe that QUATTRO PRO's analytical and presentation abilities have the power to revolutionize education. But why not find out for yourself?

Simply complete and return the enclosed Instructor Discount Coupon, or call 1-800-331-0877. You'll receive the complete $495 program—including our risk-free guarantee—for just $69.95. All we ask is that we receive your order before October 31, 1990.

Don't miss this once-in-a-lifetime chance to put you and your students on the leading edge of spreadsheet technology for only $69.95 Order your specially priced copy of QUATTRO PRO today.

Cordially,

/s/ PHILIPPE KAHN
Philippe Kahn
President, Borland International, Inc.

P.S. Remember, Borland is backing this offer with a 60-day money-back guarantee. If for any reason QUATTRO PRO doesn't completely satisfy you call us and we'll arrange a full refund. You risk absolutely nothing and have so much to gain!

PLAINTIFF'S EXHIBIT 32

Date: Fri Jul 31, 1992 5:42 pm

TO: Dave Anderson / 350-9162
TO: John Santee / 436-7190
CC: Robert Warfield / 359-9158
CC: Keith Davis / 424-9795
CC: Tom Burbage / 421-7558

Subject: CR4314

The following is a Deferred bug with CROM. Do we need to fix it before the new Quattro Pro 4.0 disk set is created?

CR4314—New Crom feature have not been added to the "phantom" 123 menu attached to QUATTRO.MU. Macros that access new Crom features run properly using 123.MU, but not with QUATTRO.MU loaded and <Tools><Macro><Keyreader> set to <Yes>.

PLAINTIFF'S EXHIBIT 33

CONFIDENTIAL
Internal Bug Report

This Information is for Borland International Internal Use Only

Product: Quattro Pro 4.0i (CROM)

Program Area:	Macro
Keyword:	Key Reader
Found By:	Tom Burbage
Submitted By:	Tom Burbage
Reported From:	QA
Date Reported:	2/12/92
Best:	No
IBR Number:	CR 4314
Status:	Open
Category:	C
Priority:	2
Version #:	4
Build #:	Build .147
Attachments:	None
Tech Note:	No

Brief Description:

New Crom features have not been added to the "phantom" 123 menu attached to QUATTRO.MU. Macros that access new Crom features run properly using 123.MU, but not with QUATTRO.MU loaded and <Tools><Macro><KeyReader> set to <Yes>. Example below.

Steps to Reproduce

1) A1: '/wgdpg
2) /<Options><Startup><MenuTree>123
3) <Shift-F2><Alt-F2>A1<Enter> ;<SpaceBar> to step through; <Break>

4) /<Worksheet><Global><Default><Files><MenuTree> QUATTRO
5) /<Tools><Macro><KeyReader><Yes>
6) Repeat step 3

Resolution Information

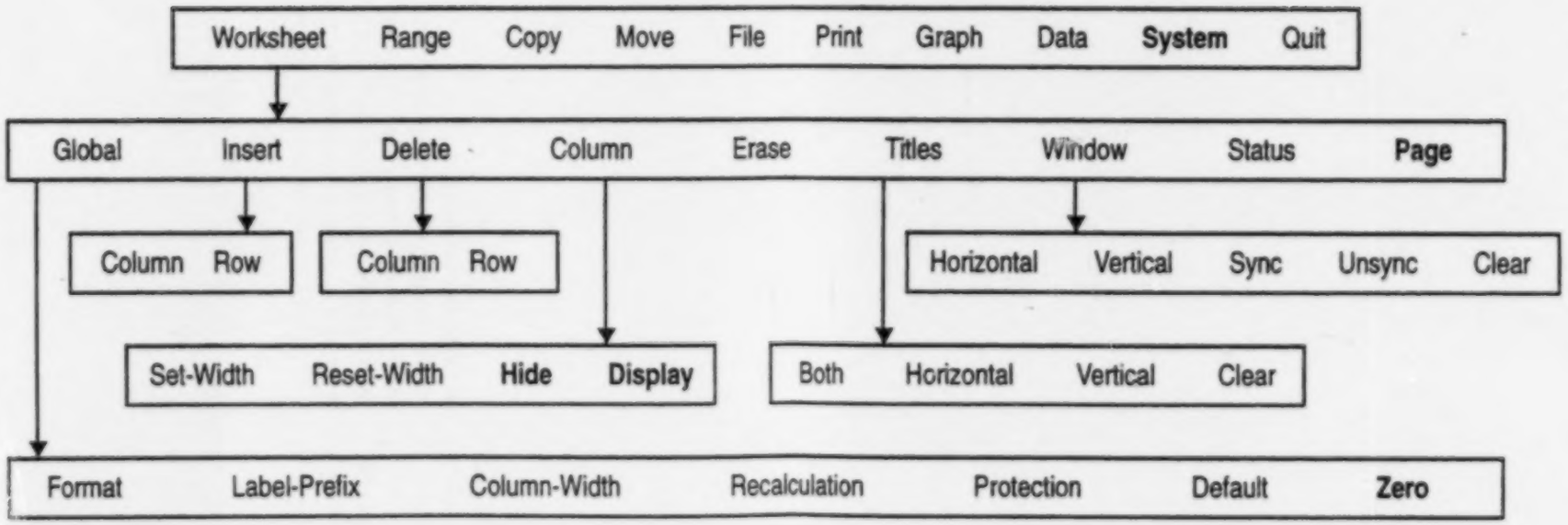
Resolution:
Programmer:

Verification:
QA Engineer:

Resolved in Build:
Date Resolved:

Verified in Build:
Date Verified:

LOTUS COMMAND TREE
(Portion of P. 139 of Plaintiff's Trial Exhibit No. 36, Dkt. No. 359;
Dybvig, Philip H., The Lotus Tutorial)





Four out of five 1-2-3 users prefer Quattro Pro

When a group of Lotus 1-2-3 users compared Quattro® Pro 4.0 and 1-2-3 side-by-side, they made a startling discovery: four out of five Lotus users preferred Quattro Pro. They found it easier to use, richer in features, and more productive. The study was done by Usability Sciences Corporation, a highly regarded independent testing laboratory. (Coincidentally, this is the same company Lotus uses to do its usability studies.)

Borland dumps old Lotus menus, keeps macro compatibility

About 90% of Quattro Pro users use only the Quattro Pro native menus. They no longer use the old Lotus slash key. And it's now obvious that Lotus doesn't want other companies supporting its outdated user interface either. Therefore, Borland has decided to give users what they prefer. Borland is now shipping Quattro Pro 4.0 with only the more efficient native menus—the ones users love. And Quattro Pro still has 1-2-3 file, publishing style, and macro compatibility.

The final mistake: Lotus goes it alone

Lotus' latest strategy of closing the menu-command standard is the final blow to users. After all, what is Lotus known for? *The best technology?* Certainly not. *The best spreadsheet?* Not even close. *Litigation rather than innovation?* Absolutely. And now Lotus is on the verge of losing its largest asset: users who have invested in the Lotus

Quattro Pro 4.0 is . . .

". . . the best DOS spreadsheet you can buy."
PC Computing, March 1992

". . . the ultimate DOS spreadsheet."
PC Magazine, April 28, 1992

". . . the best non-Windows spreadsheet . . ."
PC World, March 1992

". . . the last DOS spreadsheet ever."
Office Technology Management, February 1992

"The best DOS spreadsheet in the world."
PC User (U.K.), March 24, 1992

menu standard. By making its menu system a closed standard, Lotus is not the only loser. You lose, because Lotus is apparently trying to eliminate your right to choose. Fortunately, you don't lose much, because 1-2-3 users prefer Quattro Pro's better design and Quattro Pro can still run their 1-2-3 macros.

Lotus closes standard, loses the war

So Lotus may have won a small battle by keeping a menu system to themselves, but they'll lose a very big war with their Berlin Wall customer strategy. Borland, on the other hand, has always believed that open standards promote competition, spur innovation, and grow markets.

Lotus and Ashton-Tate: history repeats itself

Ashton-Tate® tried to establish a closed standard too. They tried to force users into their camp and to keep out the competition through litigation. Borland's acquisition of Ashton-Tate reversed this. Borland dropped Ashton-Tate's lawsuit against Fox and paved the way for competing products from several companies. Customers have won, because they can freely choose other products compatible with the open dBASE® standard.

Quattro Pro: the best DOS spreadsheet

Quattro Pro has won 42 awards, dwarfing Lotus in side-by-side comparisons. And Quattro Pro 4.0 has been called the best DOS spreadsheet by leading computer trade publications. It's proof that competition works.

You be the judge

For a good lawyer, call Lotus' law firm. For the best DOS spreadsheet, get Quattro Pro.

See your dealer today or call Borland at 1-800-331-0877, ext. 5725



BORLAND

Software Craftsmanship

Copyright © 1992 Borland International, Inc. All rights reserved. Quattro Pro, dBASE, and Ashton-Tate are registered trademarks of Borland International, Inc. BI 4529

J.A. 919

DEFENDANT'S EXHIBIT 505

THERE'S NOTHING INNOVATIVE ABOUT COPYING.

On Friday, July 31, 1992, a U.S. District Court ruled that Borland's Quattro® and Quattro Pro® spreadsheets infringe the copyrights of Lotus® 1-2-3®.

In its ruling, the Court concluded that ". . . the Quattro programs derive from illicit copying," holding that "Lotus has sued" and "Borland is liable."

Lotus innovated. Borland copied.

We sued to protect our intellectual property rights. And in winning we've helped preserve an environment in which independent software developers can freely develop innovative new products without fear that their creative work will be stolen. We sued to protect innovation, not to stifle it. Borland's copying is no different than someone plagiarizing *The Grapes of Wrath*, changing the ending, and calling it a new novel. It's really that simple.

But the courtroom is just one of the places where Borland® has lost.

They've lost in the marketplace, where 1-2-3 for DOS continues to dominate. Based on a recent report from the Software Publishers Association,® Lotus 1-2-3 accounts for seven out of 10 new DOS spreadsheet purchases and over 80% of all DOS spreadsheet revenue.

They've lost in performance categories, despite their claims to the contrary. For example, in a recent National Software Testing Laboratories performance test,** 1-2-3 Release 2.4 beats Quattro Pro 4.0 hands down in areas users care about most. The tests reveal that 1-2-3 Release 2.4 is four times faster than Quattro Pro 4.0 in retrieving a .WK1 file; 18 times faster at moving a block of data; three times faster at product load; one-and-a-half times faster at printing; and 35 times faster scrolling right or left in graphical mode.

They've lost on the innovation front, too. Lotus 1-2-3 was the first DOS spreadsheet to introduce innovative technologies such as true 3D worksheets, the Viewer, WYSIWYG display; DataLens®, and unique one-click SmartIcons. And we're still the only spreadsheet that offers true compatibility across platforms.

Who should you trust?

But perhaps most importantly, Borland lost what matters most to customers: credibility.

For instance, Borland told the Court they needed to copy our menus to achieve macro compatibility with 1-2-3. Now they tell their customers that the 1-2-3 menus aren't critical to compatibility.

So ask yourself: to what extent can you trust a company that values what is expedient over what is legal? And to what extent can you rely on the product it wants you to buy?

Here's our advice: choose the product, and the company, you can trust. Choose Lotus. After all, we're the best in the business at building innovative spreadsheets. Always have been, always will be.

Case closed.

LOTUS (logo)

DEFENDANT'S EXHIBIT 507

Jim Manzi

New York PC Users Group
October 17, 1990

* * *

[7] First, we think you'll find that Release 3.1 is actually a more graphical spreadsheet than either Excel or Quattro Pro. This, you're probably thinking, is a strong statement, coming from [the company that owns 70 percent of the DOS, character-based spreadsheet market.]*

* * *

---

* Received only for bracketed phrase on p. 7.

J.A. 923

DEFENDANT'S EXHIBIT 508

SALES SPEECH

Happy New Year!

Did everyone get a copy of the Lotus Game? It's just like real life. If you can ship and sell enough new products—without being distracted by little things like development disasters—you too can retire young to the paradise of your choice.

You can tell the game was designed in Cambridge; it has too much emphasis on reorgs. We've found that no one actually plays the game—they just read the cards. There were a few cards missing from the final version, though. Vince Kelley didn't want everyone in the company to know what's really going on out here in (Los Angeles, Chicago, New York).

But seriously, you can't hide the kind of success we had in 1987. We shipped our three millionth spreadsheet, less than one year after we shipped our two millionth in October 1986. Worldwide sales are up nearly 30 percent in the first three quarters of 1988. 1-2-3 and Symphony are still the corporate standards around the world. The proof of that is in Japan, where 1-2-3 shot to the top of the charts and stayed there.

The funny thing is that a year ago we predicted that spreadsheet sales would be flat. The product segment seemed to be mature, and we assumed that sales of a product as old as 1-2-3 would start to decline.

Were we ever wrong about that! Our performance shows that either Steve Crummey was sandbagging, or we don't know how to use our own products very well. Amazingly enough, sales haven't even peaked yet. Witness the fact that we had our largest shipment ever of 1-2-3 in our most recent quarter. We also had our best quarter for Symphony sales. We've either got the best products on the market or the best sales force in the industry. Maybe we've got both.

We also got a boost from another of our major accomplishments—our strategic agreement with IBM. It's funny, but before April, Wall Street analysts thought Microsoft had the inside track. The analysts who follow Lotus worried about Microsoft coming to market with Excel. And they worried that someday maybe IBM. would get into the spreadsheet market, and then all bets on Lotus would be off.

Well, IBM. did get into the spreadsheet market—by selling Lotus products. After April, the playing field was level. The agreement removed not only a potential competitor, but the reason why our stock was discounted. It changed a potential rival into an ally, overnight.

And I do mean overnight. We really caught the analysts by surprise. When Jim Manzi made the announcement at Lotus Week, I saw seven Wall Street analysts leap out of their seats and rush for the phone!

As you know, the IBM. agreement gives us some very terrific benefits. It gives us an entree into the IBM. market. 1-2-3/M runs on everything from 3090 mainframes to the new 9370 mid-size machine. It extends 1-2-3 to 3270 terminal users. All this is nice, but what's really important is that the agreement strategically reinforces 1-2-3 as the corporate spreadsheet standard. People using 1-2-3 on PCs can now roll their data over to a mainframe that can consolidate spreadsheets from hundreds of departments into one giant spreadsheet. And that is more than any of our competitors can do.

We've had another major success story with graphics. Two years ago, at the beginning of 1986, DRI was an independent company making an extremely popular graphics package called Chartmaster. And GCI was an independent company making Freelance, a graphics package that ran a distant second to Chartmaster. That started to change when Ashton-Tate bought DRI, and Lotus bought GCI.

Common wisdom had it that Ashton-Tate made the better deal. After all, they bought the market leader. But Chart-

master. didn't seem to fit into their strategy. Plus, Ashton-Tate moved DRI from Massachusetts to California, and in the process conducted a massive shakeout. When it was over, only a few DRI people were left.

In the meanwhile, over at Lotus, we were convinced that Freelance fit into our strategy. We bought GCI because we believed Freelance was the best product on the market, and because it works with 1-2-3. We worked hard to integrate GCI and its people into our organization.

I think we know now who really made the better deal. The situation of the two products is completely reversed. Freelance Plus has a two-to-one lead and holds 48 percent of the graphics software market. Chartmaster. lags well behind with a market share between 20 and 30 percent.

Freelance has given us a healthy infusion of revenues. And its success will continue to build. With the introduction of Graphwriter 2, we expect similar spectacular results.

We also established market leadership in Information Services in 1987. We introduced One Source in the fall of 1986 and shipped in the spring of 1987. One Source quickly made us the leader in CD-ROM services, nailing the numbers side of that market. And with the completion of our acquisition of Datext this past fall, we're going to nail down the text side of the market. With Datext we've gotten an interesting product and a lot of good people, including their sales force.

So we've essentially doubled our position in the market and now offer both numbers and text. That's important, because when some people are doing a search on an annual report they want the numbers, while others want to see the company's statement of operations. Together, One Source and Datext give us a tremendous one-two punch in the delivery of financial information on CD-ROM.

According to 3M, which supplies us—and our competitors—with CD-ROMs, we now own 80 percent of the market. The

breadth of our offerings and our market dominance give us immense leverage, both with our customers and with other suppliers of information. Our experience in creating and delivering weekly updates is just as important. It was no small feat to create our own distribution channel, and this channel will give us leverage down the road.

While we've been making advances all along the line on the product front, we've also been strengthening our ties to corporate customers. We took an asset we built up during 1985 and 1986—you, our direct sales force—and used it to develop relationships with corporate accounts in 1987. We got our largest customers together at the Lotus Forum to brief them on the direction we're taking. We put together account teams to service these customers. We started a Product Council composed of corporate users, briefed them on our product plans, and used their feedback to help develop Release 3 of 1-2-3. We went on a 13-city tour that updated 2,000 customers on what's happening at Lotus. We have a Design Partners program, an Application Partners program, Corporate Councils, and Alpha workshops. All of these relationship building activities puts us two years ahead of our customers.

In 1987 we also continued our geographic expansion. In the United States we expanded our office space in eight cities. On the international fronts we opened new sales offices in Toronto, Sydney, Stockholm and Milan.

Finally, throughout the company, we have significantly increased the depth and breadth of our management talent. We have created a structure that fosters an appropriate balance between centralized and decentralized decision making. And we are creating a risk-taking, policy-making level of middle managers who will help define what a $500 million or a $1 billion software company should be and who will help chart the course to get there.

What has all this activity accomplished? It's given us a way to communicate our messages directly to our customers, with

no intermediaries. Unlike our competition, we don't have to rely on the trade press or dealers to convey our messages to end users. Some of our competitors now are trying to develop a direct sales force, but we have a two-year lead. And when you consider the fear, uncertainty, and doubt that surround the purchase of software, that lead gives us a tremendous advantage. You are the ones that are right there to help our customers make the right decision—the Lotus decision.

We did a lot of things right in 1987, but we also learned some tough lessons along the way. We found out that the engineering and scientific market wasn't as cohesive thought it would be. We've decided that we can't doing anything more effectively in this market, and we're getting out.

Selling low-priced products through the dealer channel wasn't as easy as we thought it would be. It's hard to get dealers excited about selling a product when the margin is low. So we made the decision to sell small products by catalog, the way we sell Lotus Select, and they're doing fine.

And getting T-A-C and the Networker to market took longer than we thought it would. The good news for 1988 is that we've finally shipped Networker. The better news is that with Release 3 of 1-2-3, Lotus DBMS, and a number of other things we have up our sleeve, our network strategy should emerge and unfold in 1988.

We'll need to remember the lessons we've learned—from both our successes and our mistakes—to deal successfully with the challenges we face in 1988. We made a lot of promises to our customers about product development that we now have to fulfill. So our first and foremost challenge is to help hundreds of thousands of customers through major upgrades on most of our product lines. It will be a lot of work, but we should handle the transitions a lot more smoothly than we handled the one from 1-2-3 to Release 2 of 1-2-3. If we do this right, we'll go home a lot richer and hap-

pier. If we mess up—well, I hear there's a sales territory open in Death Valley.

Our first test will come with Symphony 2.0, which we will formally announce on Thursday. The next—and crucial—test will come in the second quarter with Release 3 of 1-2-3. And let's not forget the upgrades from Jazz to Modern Jazz, Manuscript 1.0 to 2.0, Metro 1.0 to 1.1 in the first quarter, and other products later in the year.

It looks like we'll remember 1988 as the year of the upgrade.

The second challenge we face this year is to maintain our share of the spreadsheet market against increased competition. PC Excel is hitting us at the high end of the spreadsheet market, while Quattro is hitting us at the low end. And there are others out there waiting in the wings. The challenge will be to maintain our high standards of business ethics—and not get suckered into playing by someone else's rules. We'll continue to emphasize the benefits of our products and product strategy, rather than knock the competition.

We'll be giving you briefings later in January and February to remind you of our competitive advantages. But for now, concentrate on the larger issue. We already own 70 percent of the market share, and we have to take care of it. If at the end of 1988 we still own that 70 percent, we're home free. Any increases in market share are pure gravy. But if we allow ourselves to be distracted by worrying about the 30 percent share we don't have—by trying to chase incremental increases in market share—we could lose some of what we already have.

Frankly, we think that if Excel and Quattro take market share, it will be from the clones. So we won't try to stop them from selling their products—we'll just let them fight over the 30 percent of the market that are not Lotus users. Let's not worry about what's happening on the fringes.

Our third challenge is to establish a leadership position in new market segments and in new product categories. This

includes new platforms for 1-2-3, such as Macs, mainframes, and the Presentation Manager. It includes new markets for Lotus—like he database market—with Lotus DBMS. It also includes entirely new categories—for example Personal Information Management. All of these are important new areas for the future growth of the company.

Finally, our fourth challenge will be to develop an active partnership program with our distribution channel. We need dealers and distributors to support spreadsheet sales aggressively. We need to use the distribution channel more intelligently by developing differentiated marketing programs by type of channel specialty. We need to differentiate Lotus based on our ability to help the channel expand sales and profit opportunities. And we need to offset Microsoft's 1-2-3-targeted profit incentives.

So what's our strategy for attaining our goals and meeting our challenges? If this slide looks surprisingly like the last slide, it's because our strategy in 1988 is to meet our challenges head on. We'll focus on helping our customers get through the upgrade process as painlessly as possible. We intend to maintain our share of the spreadsheet market. We expect to establish leadership positions in new segments and new categories. And we hope to develop an active partnership program with the distribution channel.

To put it even more succinctly, our strategy is to do two things right. The first is to protect the investment our customers have made in our products. We'll do this by providing them with smooth upgrades to new technologies and win the best sales, service, and support possible. The second is to continue to find innovative ways to make PC technology more useful to hundreds of thousands of people. No customer should have to sacrifice the technology it already has to take advantage of new technologies.

Actually,—as all of you know—coming up with the strategy is the easy part. Working out the tactics is the hard part. And

implementing them is the hardest part of all. That's up to you. We count on you—the face that the customer sees most often—to make our strategy a success.

This makes a wonderful segue to our next speaker, who'll talk to you about implementation. But before I conclude, let me convey one final message.

On behalf of top management and all the people in Cambridge, thank you for your outstanding efforts in 1987. As Jim Manzi has often said, you are the most talented and professional sales organization in the industry. And thank you, in advance, for the outstanding efforts we expect you'll continue to make in 1988. It puts food on the plate for all of us.

EXHIBIT 517

```
(
 "" PICKLETTER "W"          Quit  ?=592
    ""
 (
  "" PICKLETTER "G"         Quit  ?=591
     ""
  (
   "" PICKLETTER "F"        Quit  ?=591  (/ Defaults:Format)
      ""
   "" PICKLETTER "L"        Quit  ?=591  (/ Defaults:Alignment)
      ""
   "" PICKLETTER "C"        Quit  ?=591  (/ Defaults:ColWidth)
      ""
   "" PICKLETTER "R"        Quit  ?=591
      ""
   (
    "" PICKLETTER "N"       Quit  ?=591  (/ CompCalc:Natural)
       ""
    "" PICKLETTER "C"       Quit  ?=591  (/ CompCalc:ColWise)
       ""
    "" PICKLETTER "R"       Quit  ?=591  (/ CompCalc:RowWise)
       ""
    "" PICKLETTER "A"       Quit  ?=591  (/ CompCalc:Automatic)
       ""
    "" PICKLETTER "M"       Quit  ?=591  (/ CompCalc:Manual)
       ""
    "" PICKLETTER "B"       Quit  ?=591  (/ CompCalc:Background)
       ""
    "" PICKLETTER "I"       Quit  ?=591  (/ Defaults:RecalcIteration)
   )
   "" PICKLETTER "P"        Quit  ?=591
      ""
   (
    "" PICKLETTER "E"       Quit  ?=591  (/ Protection:Enable)
       ""
    "" PICKLETTER "D"       Quit  ?=591  (/ Protection:Disable)
       ""
   )
   "" PICKLETTER "D"        Quit  ?=590
      ""
   (
    "" PICKLETTER "P"       Stay  ?=97
       ""
    (
     "" PICKLETTER "I"      Stay  ?=96   (/ Hardware:LotusDevice)
        ""
     "" PICKLETTER "A"      Stay  ?=94   (/ Hardwr  :AutoLf)
        ""
     "" PICKLETTER "L"      Stay  ?=210  (/ Defaults:LeftMargin)
        ""
     "" PICKLETTER "R"      Stay  ?=210  (/ Defaults:RightMargin)
        ""
     "" PICKLETTER "T"      Stay  ?=210  (/ Defaults:TopMargin)
```

```
-
-- PICKLETTER "B"        Stay  ?=210  (/ Defaults;BottomMargin)
-
-- PICKLETTER "P"        Stay  ?=210  (/ Defaults;PageLength)
-
-- PICKLETTER "W"        Stay  ?=94   (/ Hardware;SingleSheet)
-
-- PICKLETTER "S"        Stay  ?=206  (/ Defaults;Setup)
-
-- PICKLETTER "N"        Stay  ?=95   (/ Defaults;PrinterName)
-
-- PICKLETTER "Q"        Quit  ?=97
-
)
-- PICKLETTER "D"        Stay  ?=81   (/ Defaults;Directory)
-
-- PICKLETTER "S"        Stay  ?=590
-
(
-- PICKLETTER "I"        Quit  ?=0    (/ Hardware;LotusDevice)
-
-- PICKLETTER "A"        Quit  ?=0    (/ Hardware;AutoLf)
-
-- PICKLETTER "L"        Quit  ?=0    (/ Defaults;LeftMargin)
-
-- PICKLETTER "R"        Quit  ?=0    (/ Defaults;RightMargin)
-
-- PICKLETTER "T"        Quit  ?=0    (/ Defaults;TopMargin)
-
-- PICKLETTER "B"        Quit  ?=0    (/ Defaults;BottomMargin)
-
-- PICKLETTER "P"        Quit  ?=0    (/ Defaults;PageLength)
-
-- PICKLETTER "W"        Quit  ?=0    (/ Hardware;SingleSheet)
-
-- PICKLETTER "S"        Quit  ?=0    (/ Defaults;Setup)
-
-- PICKLETTER "N"        Quit  ?=0    (/ Defaults;PrinterName)
-
-- PICKLETTER "D"        Quit  ?=0    (/ Defaults;Directory)
-
-- PICKLETTER "C"        Quit  ?=0    (/ Defaults;ClockFormat)
-
-- PICKLETTER "P"        Quit  ?=0    (/ Intnl;Punctuation)
-
-- PICKLETTER "C"        Quit  ?=0    (/ Intnl;Currency)
-
-- PICKLETTER "D"        Quit  ?=0    (/ FormatChanges;IntlDate)
-
-- PICKLETTER "T"        Quit  ?=0    (/ FormatChanges;IntlTime)
-
)
-- PICKLETTER "U"        Stay  ?=62   (/ Defaults;UpdateAll)
-
```



```
-- PICKLETTER "O"        Stay  ?=589
-
(
-- PICKLETTER "I"        Quit  ?=84
-
(
-- PICKLETTER "P"        Stay  ?=84   (/ Intnl;Punctuation)
-
-- PICKLETTER "C"        Stay  ?=84   (/ Intnl;Currency)
-
-- PICKLETTER "D"        Stay  ?=84   (/ FormatChanges;IntlDate)
-
-- PICKLETTER "T"        Stay  ?=84   (/ FormatChanges;IntlTime)
-
-- PICKLETTER "S"        Stay  ?=84   (/ Intnl;UseSortTable)
-
-- PICKLETTER "L"        Stay  ?=84   (/ Intnl;LICS)
-
-- PICKLETTER "O"        Stay  ?=84   (/ Intnl;PrintComposed)
-
-- PICKLETTER "Q"        Quit  ?=589
-
)
-- PICKLETTER "H"        Quit  ?=589  (/ Defaults;HelpAccess)
-
-- PICKLETTER "C"        Quit  ?=63   (/ Defaults;ClockFormat)
-
-                  Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "U"        Quit  ?=63   (/ Defaults;Undo)
-
-- PICKLETTER "M"        Quit  ?=63   (/ Defaults;Suppress)
-
-- PICKLETTER "E"        Quit  ?=63   (/ Defaults;ExpMem)
-
-"- PICKLETTER "B"       Quit  ?=77   (/ Startup;Beep)
-
-- PICKLETTER "P"    -   Quit  ?=63
-
(
-- PICKLETTER "N"        Stay  ?=63   (/ Paradox;NetType)
-
-- PICKLETTER "D"        Stay  ?=63   (/ Paradox;NetDir)
-
-- PICKLETTER "R"        Stay  ?=63   (/ Paradox;NetRetries)
-
-- PICKLETTER "Q"        Quit  ?=63
-
)
)
-                 Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "H"        Stay  ?=102
-
```

```
(
 PICKLETTER "S"        Quit  ?=101
(
 PICKLETTER "S"        Stay  ?=101  (/ ScreenHardware;GraphScreenType)
 PICKLETTER "R"        Stay  ?=100  (/ Graph;ScreenMode)
 PICKLETTER "A"        Stay  ?=99   (/ ScreenHardware;AspectRatio)
 PICKLETTER "C"        Stay  ?=98   (/ ScreenHardware;Retrace)
 PICKLETTER "Q"        Quit  ?=101
)
 PICKLETTER "P"        Quit  ?=97
(
 PICKLETTER "1"        Stay  ?=96
(
 PICKLETTER "T"        Stay  ?=96   (/ GPrinter1;Type)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter1;ShowMake)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter1;ShowModel)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter1;ShowMode)
 PICKLETTER "D"        Stay  ?=96   (/ GPrinter1;Device)
 PICKLETTER "B"        Stay  ?=96   (/ GPrinter1;Baud)
 PICKLETTER "P"        Stay  ?=96   (/ GPrinter1;Parity)
 PICKLETTER "S"        Stay  ?=96   (/ GPrinter1;Stop)
 PICKLETTER "Q"        Quit  ?=96
)
 PICKLETTER "2"        Stay  ?=96
(
 PICKLETTER "T"        Stay  ?=96   (/ GPrinter2;Type)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter2;ShowMake)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter2;ShowModel)
 PICKLETTER "M"        Stay  ?=0    (/ GPrinter2;ShowMode)
 PICKLETTER "D"        Stay  ?=96   (/ GPrinter2;Device)
 PICKLETTER "B"        Stay  ?=96   (/ GPrinter2;Baud)
```



```
 PICKLETTER "P"        Stay  ?=96   (/ GPrinter2;Parity)
 PICKLETTER "S"        Stay  ?=96   (/ GPrinter2;Stop)
 PICKLETTER "Q"        Quit  ?=96
)
 PICKLETTER "D"        Stay  ?=95   (/ Defaults;PrinterName)
               Stay  ?=0
  DISPLAY-ONLY
 PICKLETTER "P"        Stay  ?=94   (/ GraphPrint;PlotSpeed)
 PICKLETTER "F"        Stay  ?=93
(
 PICKLETTER "L"        Quit  ?=93
(
 PICKLETTER "L"        Quit  ?=93   (/ Hardware;LJetLeft)
 PICKLETTER "R"        Quit  ?=93   (/ Hardware;LJetRight)
)
 PICKLETTER "A"        Quit  ?=93   (/ Hardware;AutoFonts)
)
 PICKLETTER "A"        Stay  ?=94   (/ Hardware;AutoLf)
 PICKLETTER "S"        Stay  ?=94   (/ Hardware;SingleSheet)
 PICKLETTER "Q"        Quit  ?=97
)
 PICKLETTER "N"        Quit  ?=0    (/ Basics;ShowMem)
 PICKLETTER "E"        Quit  ?=0    (/ Basics;ShowEMS)
 PICKLETTER "C"        Quit  ?=0    (/ Basics;ShowCoProc)
)
 PICKLETTER "C"        Stay  ?=92
(
 PICKLETTER "M"        Stay  ?=91
(
 PICKLETTER "F"        Stay  ?=91   (/ MenuColors;Frame)
 PICKLETTER "B"        Stay  ?=91   (/ MenuColors;Banner)
 PICKLETTER "T"        Stay  ?=91   (/ MenuColors;Text)
```

```
" PICKLETTER "K"        Stay  ?=91   {/ MenuColors;FirstLetter}
"
" PICKLETTER "H"        Stay  ?=91   {/ MenuColors;MenuBar}
"
" PICKLETTER "S"        Stay  ?=91   {/ MenuColors;Settings}
"
" PICKLETTER "E"        Stay  ?=91   {/ MenuColors;Explanation}
"
" PICKLETTER "D"        Stay  ?=91   {/ Startup;Shadow}
"
" PICKLETTER "M"        Stay  ?=91   {/ Startup;PaletteCol}
"
" PICKLETTER "F"        Stay  ?=0
"
" PICKLETTER "S"        Stay  ?=91   {/ Startup;ShadowChar}
"
" PICKLETTER "Q"        Quit  ?=91
"
)
" PICKLETTER "D"        Stay  ?=90
"
(
" PICKLETTER "C"        Stay  ?=0
"
" PICKLETTER "S"        Stay  ?=90   {/ Color;Status}
"
" PICKLETTER "H"        Stay  ?=90   {/ Color;Indicators}
"
" PICKLETTER "E"        Stay  ?=90   {/ ErrorColor;SetErrorColor}
"
" PICKLETTER "B"        Stay  ?=90   {/ Startup;DesktopColor}
"
" PICKLETTER "F"        Stay  ?=0
"
" PICKLETTER "D"        Stay  ?=90   {/ Startup;DesktopChar}
"
" PICKLETTER "Q"        Quit  ?=90
"
)
" PICKLETTER "S"        Stay  ?=89
"
(
" PICKLETTER "F"        Stay  ?=89   {/ Color;Frame}
"
" PICKLETTER "B"        Stay  ?=89   {/ Color;Banner}
"
" PICKLETTER "C"        Stay  ?=89   {/ Color;Cells}
"
" PICKLETTER "B"        Stay  ?=89   {/ Color;Border}
"
" PICKLETTER "T"        Stay  ?=89   {/ Color;Titles}
"
" PICKLETTER "H"        Stay  ?=89   {/ Color;Cursor}
"
```



```
" PICKLETTER "G"        Stay  ?=89   {/ Color;GraphFrame}
"
" PICKLETTER "I"        Stay  ?=89   {/ Color;Edit}
"
" PICKLETTER "U"        Stay  ?=89   {/ Color;Unprotect}
"
" PICKLETTER "L"        Stay  ?=89   {/ ValueColors;Labels}
"
" PICKLETTER "S"        Stay  ?=89   {/ Color;Shading}
"
" PICKLETTER "D"        Stay  ?=89   {/ Color;LineDrawing}
"
" PICKLETTER "Q"        Quit  ?=89
"
)
" PICKLETTER "C"        Stay  ?=88
"
(
" PICKLETTER "O"        Stay  ?=88   {/ ValueColors;Enable}
"
" PICKLETTER "E"        Stay  ?=88   {/ ValueColors;Err}
"
" PICKLETTER "S"        Stay  ?=88   {/ ValueColors;Min}
"
" PICKLETTER "G"        Stay  ?=88   {/ ValueColors;Max}
"
" PICKLETTER "B"        Stay  ?=88   {/ ValueColors;Low}
"
" PICKLETTER "N"        Stay  ?=88   {/ ValueColors;Normal}
"
" PICKLETTER "A"        Stay  ?=88   {/ ValueColors;High}
"
" PICKLETTER "Q"        Quit  ?=88
"
)
" PICKLETTER "H"        Stay  ?=87
"
(
" PICKLETTER "F"        Stay  ?=87   {/ HelpColors;Frame}
"
" PICKLETTER "B"        Stay  ?=87   {/ HelpColors;Banner}
"
" PICKLETTER "T"        Stay  ?=87   {/ HelpColors;Text}
"
" PICKLETTER "K"        Stay  ?=87   {/ HelpColors;Keyword}
"
" PICKLETTER "H"        Stay  ?=87   {/ HelpColors;Highlight}
"
" PICKLETTER "Q"        Quit  ?=87
"
)
" PICKLETTER "F"        Stay  ?=86
"
(
```

```
"PICKLETTER "F"      Stay  ?=86  (/ FileMgrColors;Frame)
"PICKLETTER "B"      Stay  ?=86  (/ FileMgrColors;Banner)
"PICKLETTER "T"      Stay  ?=86  (/ FileMgrColors;Text)
"PICKLETTER "A"      Stay  ?=86  (/ FileMgrColors;ActiveCursor)
"PICKLETTER "I"      Stay  ?=86  (/ FileMgrColors;InactiveCursor)
"PICKLETTER "M"      Stay  ?=86  (/ FileMgrColors;Marked)
"PICKLETTER "C"      Stay  ?=86  (/ FileMgrColors;Cut)
"PICKLETTER "C"      Stay  ?=86  (/ FileMgrColors;Copy)
"PICKLETTER "Q"      Quit  ?=86
)
"PICKLETTER "P"      Stay  ?=85
(
"PICKLETTER "C"      Stay  ?=85  (/ Color;ColorPalette)
"PICKLETTER "M"      Stay  ?=85  (/ Color;BWPalette)
"PICKLETTER "B"      Stay  ?=85  (/ Color;BWCGAPalette)
"PICKLETTER "Q"      Quit  ?=85
)
"PICKLETTER "Q"      Quit  ?=92
)
"PICKLETTER "M"      Stay  ?=65
(
"PICKLETTER "1"      Stay  ?=65  (/ Defaults;Button1)
"PICKLETTER "2"      Stay  ?=65  (/ Defaults;Button2)
"PICKLETTER "3"      Stay  ?=65  (/ Defaults;Button3)
"PICKLETTER "4"      Stay  ?=65  (/ Defaults;Button4)
"PICKLETTER "5"      Stay  ?=65  (/ Defaults;Button5)
"PICKLETTER "6"      Stay  ?=65  (/ Defaults;Button6)
"PICKLETTER "7"      Stay  ?=65  (/ Defaults;Button7)
"PICKLETTER "Q"      Quit  ?=65
)
```



```
"PICKLETTER "F"      Stay  ?=588
(
"PICKLETTER "A"      Quit  ?=80  (/ Startup;File)
"PICKLETTER "S"      Quit  ?=79  (/ Startup;Macro)
"PICKLETTER "F"      Quit  ?=78  (/ Startup;Extension)
"PICKLETTER "M"      Quit  ?=76  (/ Startup;Menus)
"PICKLETTER "E"      Quit  ?=71  (/ MenuBuilder;Run)
)
"PICKLETTER "G"      Stay  ?=64  (/ Defaults;GraphicsQuality)
"PICKLETTER "Q"      Quit  ?=596
)
"PICKLETTER "Z"      Quit  ?=591  (/ Defaults;Zero)
)
"PICKLETTER "I"      Quit  ?=592
(
"PICKLETTER "C"      Quit  ?=592  (/ Column;Insert)
"PICKLETTER "R"      Quit  ?=592  (/ Row;Insert)
)
"PICKLETTER "D"      Quit  ?=592
(
"PICKLETTER "C"      Quit  ?=592  (/ Column;Delete)
"PICKLETTER "R"      Quit  ?=592  (/ Row;Delete)
)
"PICKLETTER "C"      Quit  ?=592
(
"PICKLETTER "S"      Quit  ?=587  (/ Column;Width)
"PICKLETTER "R"      Quit  ?=587  (/ Column;Reset)
"PICKLETTER "H"      Quit  ?=587  (/ Column;Hide)
"PICKLETTER "D"      Quit  ?=587  (/ Column;Display)
)
"PICKLETTER "E"      Quit  ?=592  (/ Basics;Erase)
"PICKLETTER "T"      Quit  ?=592
```

```
(
-- PICKLETTER "B"        Quit  ?=592  (/ Titles;Both)
   -
-- PICKLETTER "H"        Quit  ?=592  (/ Titles;Horizontal)
   -
-- PICKLETTER "V"        Quit  ?=592  (/ Titles;Vertical)
   -
-- PICKLETTER "C"        Quit  ?=592  (/ Titles;Clear)
   -
)
-- PICKLETTER "W"        Quit  ?=592
   -
(
-- PICKLETTER "H"        Quit  ?=46   (/ Windows;Horizontal)
   -
-- PICKLETTER "V"        Quit  ?=46   (/ Windows;Vertical)
   -
-- PICKLETTER "S"        Quit  ?=46   (/ Windows;Synch)
   -
-- PICKLETTER "U"        Quit  ?=46   (/ Windows;Unsynch)
   -
-- PICKLETTER "C"        Quit  ?=46   (/ Windows;Clear)
   -
-- PICKLETTER "R"        Quit  ?=46
   -
(
-- PICKLETTER "D"        Quit  ?=46   (/ Windows;RowColDisplay)
   -
-- PICKLETTER "H"        Quit  ?=46   (/ Windows;RowColHide)
   -
)
-- PICKLETTER "M"        Quit  ?=46   (/ Windows;MapView)
   -
)
-- PICKLETTER "S"        Quit  ?=592
   -
(
-- PICKLETTER "N"        Quit  ?=0    (/ Basics;ShowMem)
   -
-- PICKLETTER "E"        Quit  ?=0    (/ Basics;ShowEMS)
   -
-- PICKLETTER "C"        Quit  ?=0    (/ Basics;ShowCoProc)
   -
-- PICKLETTER "R"        Quit  ?=0    (/ Defaults;RecalcMode)
   -
-- PICKLETTER "R"        Quit  ?=0    (/ Defaults;RecalcOrder)
   -
-- PICKLETTER "I"        Quit  ?=0    (/ Defaults;RecalcIteration)
   -
-- PICKLETTER "C"        Quit  ?=0    (/ Audit;ShowCirc)
   -
-- PICKLETTER "C"        Quit  ?=0    (/ Defaults;Format)
   -
-- PICKLETTER "L"        Quit  ?=0    (/ Defaults;Alignment)
```



```
   -
-- PICKLETTER "C"        Quit  ?=0    (/ Defaults;ColWidth)
   -
-- PICKLETTER "S"        Quit  ?=0    (/ Defaults;Zero)
   -
-- PICKLETTER "P"        Quit  ?=0    (/ Protection;Status)
   -
-- PICKLETTER "M"        Quit  ?=0    (/ Macro;Library)
   -
-- PICKLETTER "U"        Quit  ?=0    (/ Defaults;Undo)
   -
)
-- PICKLETTER "P"        Quit  ?=592  (/ Print;CreatePageBreak)
   -
-                  Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "M"        Quit  ?=571  (/ Macro;Menu)
   -
-- PICKLETTER "U"        Quit  ?=339  (/ Basics;Undo)
   -
)
-- PICKLETTER "R"        Quit  ?=586
   -
(
-- PICKLETTER "F"        Quit  ?=319  (/ Block;Format)
   -
-- PICKLETTER "L"        Quit  ?=320  (/ Block;Align)
   -
-- PICKLETTER "E"        Quit  ?=340  (/ Block;Erase)
   -
-- PICKLETTER "N"        Quit  ?=336
   -
(
-- PICKLETTER "C"        Quit  ?=334  (/ Name;Create)
   -
-- PICKLETTER "D"        Quit  ?=332  (/ Name;Delete)
   -
-- PICKLETTER "L"        Quit  ?=333
   -
(
-- PICKLETTER "R"        Quit  ?=333  (/ Name;RightCreate)
   -
-- PICKLETTER "D"        Quit  ?=333  (/ Name;UnderCreate)
   -
-- PICKLETTER "L"        Quit  ?=333  (/ Name;LeftCreate)
   -
-- PICKLETTER "U"        Quit  ?=333  (/ Name;AboveCreate)
   -
)
-- PICKLETTER "R"        Quit  ?=332  (/ Name;Reset)
   -
-- PICKLETTER "T"        Quit  ?=331  (/ Name;Table)
   -
)
```

```
" PICKLETTER "J"          Quit  ?=144  (/ Block;Justify)
  -
" PICKLETTER "P"          Quit  ?=314  (/ Block;Protect)
  -
" PICKLETTER "U"          Quit  ?=314  (/ Block;Unprotect)
  -
" PICKLETTER "I"          Quit  ?=177  (/ Block;Input)
  -
" PICKLETTER "V"          Quit  ?=328  (/ Block;Values)
  -
" PICKLETTER "T"          Quit  ?=327  (/ Block;Transpose)
  -
-                   Quit  ?=0
   DISPLAY-ONLY
" PICKLETTER "C"          Quit  ?=310
  -
(
" PICKLETTER "S"          Quit  ?=309  (/ Block;SetWidth)
  -
" PICKLETTER "R"          Quit  ?=312  (/ Block;ResetWidth)
  -
" PICKLETTER "A"          Quit  ?=308  (/ Block;AdjustWidth)
  -
)
" PICKLETTER "S"          Quit  ?=326
  -
(
" PICKLETTER "B"          Stay  ?=325  (/ Audit;ReplaceRange)
  -
" PICKLETTER "S"          Stay  ?=325  (/ Audit;SearchString)
  -
" PICKLETTER "R"          Stay  ?=325  (/ Audit;ReplaceString)
  -
" PICKLETTER "O"          Stay  ?=0
  -
" PICKLETTER "L"          Stay  ?=324  (/ Audit;SearchLookin)
  -
(
" PICKLETTER "F"          Quit  ?=324  (/ Audit;SearchFormula)
  -
" PICKLETTER "V"          Quit  ?=324  (/ Audit;SearchValue)
  -
" PICKLETTER "C"          Quit  ?=324  (/ Audit;SearchCondition)
  -
)
" PICKLETTER "D"          Stay  ?=324  (/ Audit;SearchDirection)
  -
(
" PICKLETTER "R"          Quit  ?=324  (/ Audit;SearchByRow)
  -
" PICKLETTER "C"          Quit  ?=324  (/ Audit;SearchByCol)
  -
)
" PICKLETTER "M"          Stay  ?=324  (/ Audit;SearchMatch)
```



```
  -
(
" PICKLETTER "P"          Quit  ?=324  (/ Audit;SearchForPart)
  -
" PICKLETTER "W"          Quit  ?=324  (/ Audit;SearchForWhole)
  -
)
" PICKLETTER "C"          Stay  ?=323  (/ Audit;SearchCase)
  -
(
" PICKLETTER "A"          Quit  ?=323  (/ Audit;SearchAnyCase)
  -
" PICKLETTER "E"          Quit  ?=323  (/ Audit;SearchExactCase)
  -
)
" PICKLETTER "O"          Stay  ?=0    (/ Audit;SearchReset)
  -
-                   Stay  ?=0
   DISPLAY-ONLY
" PICKLETTER "N"          Stay  ?=323  (/ Audit;Replace)
  -
" PICKLETTER "P"          Stay  ?=323  (/ Audit;SearchPrev)
  -
" PICKLETTER "Q"          Quit  ?=326
  -
)
" PICKLETTER "A"          Quit  ?=320
  -
(
" PICKLETTER "G"          Quit  ?=320  (/ Publish;AlignDefault)
  -
" PICKLETTER "L"          Quit  ?=320  (/ Publish;AlignLeft)
  -
" PICKLETTER "R"          Quit  ?=320  (/ Publish;AlignRight)
  -
" PICKLETTER "C"          Quit  ?=320  (/ Publish;AlignCenter)
  -
)
" PICKLETTER "D"          Quit  ?=176
  -
(
" PICKLETTER "G"          Quit  ?=176  (/ Publish;DataEntryFormula)
  -
" PICKLETTER "L"          Quit  ?=176  (/ Publish;DataEntryLabel)
  -
" PICKLETTER "D"          Quit  ?=176  (/ Publish;DataEntryDate)
  -
)
" PICKLETTER "O"          Quit  ?=585
  -
(
" PICKLETTER "L"          Quit  ?=307  (/ Publish;LineDrawing)
  -
" PICKLETTER "S"          Quit  ?=305
```

```
    -
(
 - PICKLETTER "N"        Quit  ?=305  {/ Publish;ShadingNone}
    -
 - PICKLETTER "G"        Quit  ?=305  {/ Publish;ShadingGrey}
    -
 - PICKLETTER "B"        Quit  ?=305  {/ Publish;ShadingBlack}
    -
)
 - PICKLETTER "F"        Quit  ?=304  {/ Publish;Font}
    -
)
)
- PICKLETTER "C"         Quit  ?=348  {/ Block;Copy}
    -
- PICKLETTER "M"         Quit  ?=346  {/ Block;Move}
    -
- PICKLETTER "F"         Quit  ?=584
    -
(
 - PICKLETTER "R"        Quit  ?=386  {/ File;Retrieve}
    -
 - PICKLETTER "S"        Quit  ?=381  {/ File;Save}
    -
 - PICKLETTER "C"        Quit  ?=140
    -
(
 - PICKLETTER "C"        Quit  ?=139
    -
(
 - PICKLETTER "E"        Quit  ?=139  {/ File;CopyFile}
    -
 - PICKLETTER "N"        Quit  ?=139  {/ File;CopyRange}
    -
)
 - PICKLETTER "A"        Quit  ?=138
    -
(
 - PICKLETTER "E"        Quit  ?=138  {/ File;AddFile}
    -
 - PICKLETTER "N"        Quit  ?=138  {/ File;AddRange}
    -
)
 - PICKLETTER "S"        Quit  ?=137
    -
(
 - PICKLETTER "E"        Quit  ?=137  {/ File;SubtractFile}
    -
 - PICKLETTER "N"        Quit  ?=137  {/ File;SubtractRange}
    -
)
)
- PICKLETTER "X"         Quit  ?=136
    -
```



```
(
 - PICKLETTER "F"        Quit  ?=135  {/ File;ExtractFormulas}
    -
 - PICKLETTER "V"        Quit  ?=134  {/ File;ExtractValues}
    -
)
- PICKLETTER "E"         Quit  ?=378  {/ File;Erase}
    -
- PICKLETTER "L"         Quit  ?=584  {/ File;List}
    -
- PICKLETTER "I"         Quit  ?=143
    -
(
 - PICKLETTER "T"        Quit  ?=142  {/ File;ImportText}
    -
 - PICKLETTER "N"        Quit  ?=141  {/ File;ImportNumbers}
    -
 - PICKLETTER "C"        Quit  ?=141  {/ File;ImportComma}
    -
)
- PICKLETTER "D"         Quit  ?=377  {/ File;Directory}
    -
-                  Quit  ?=0
    DISPLAY-ONLY
- PICKLETTER "N"         Quit  ?=387  {/ View;NewWindow}
    -
- PICKLETTER "O"         Quit  ?=26   {/ View;OpenWindow}
    -
- PICKLETTER "W"         Quit  ?=22
    -
(
 - PICKLETTER "S"        Quit  ?=22   {/ System;SaveWorkspace}
    -
 - PICKLETTER "R"        Quit  ?=22   {/ System;RestoreWorkspace}
    -
)
- PICKLETTER "I"         Quit  ?=375
    -
(
 - PICKLETTER "R"        Stay  ?=374  {/ SQZ;Blanks}
    -
 - PICKLETTER "S"        Stay  ?=373  {/ SQZ;Values}
    -
 - PICKLETTER "V"        Stay  ?=372  {/ SQZ;Version}
    -
 - PICKLETTER "Q"        Quit  ?=375
    -
)
- PICKLETTER "U"         Quit  ?=0
    -
(
 - PICKLETTER "O"        Quit  ?=132  {/ Hotlink;Open}
    -
 - PICKLETTER "R"        Quit  ?=131  {/ Hotlink;Update}
```

```
-
-- PICKLETTER "C"          Quit  ?=130  (/ Hotlink;Change)
-
-- PICKLETTER "D"          Quit  ?=129  (/ Hotlink;Delete)
-
)
)
-- PICKLETTER "P"          Quit  ?=583
-
(
-- PICKLETTER "D"          Quit  ?=0
-
-- PICKLETTER "P"          Quit  ?=582  (/ Print;OutputPrinter)
-
(
-- PICKLETTER "R"          Stay  ?=582  (/ Print;Block)
-
-- PICKLETTER "L"          Stay  ?=582  (/ Print;SkipLine)
-
-- PICKLETTER "P"          Stay  ?=582  (/ Print;FormFeed)
-
-- PICKLETTER "O"          Stay  ?=582
-
(
-- PICKLETTER "H"          Stay  ?=581  (/ Print;Header)
-
-- PICKLETTER "F"          Stay  ?=581  (/ Print;Footer)
-
-- PICKLETTER "M"          Stay  ?=581
-
(
-- PICKLETTER "L"          Quit  ?=581  (/ Print;LeftMargin)
-
-- PICKLETTER "R"          Quit  ?=581  (/ Print;RightMargin)
-
-- PICKLETTER "T"          Quit  ?=581  (/ Print;TopMargin)
-
-- PICKLETTER "B"          Quit  ?=581  (/ Print;BottomMargin)
-
-                    Quit  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"          Quit  ?=581  (/ Print;Dimensions)
-
)
-- PICKLETTER "B"          Stay  ?=581
-
(
-- PICKLETTER "C"          Quit  ?=581  (/ Print;LeftBorder)
-
-- PICKLETTER "R"          Quit  ?=581  (/ Print;TopBorder)
-
)
-- PICKLETTER "S"          Stay  ?=581  (/ Print;Setup)
-
```



```
-- PICKLETTER "P"          Stay  ?=581  (/ Print;PageLength)
-
-- PICKLETTER "O"          Stay  ?=580  (/ Print;Others)
-
-                    Stay  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"          Stay  ?=581  (/ Print;Rotated)
-
-- PICKLETTER "U"          Stay  ?=581  (/ Print;Update)
-
-- PICKLETTER "Q"          Quit  ?=581
-
)
-- PICKLETTER "C"          Stay  ?=582
-
(
-- PICKLETTER "A"          Quit  ?=582  (/ Print;ResetAll)
-
-- PICKLETTER "R"          Quit  ?=582  (/ Print;ResetBlock)
-
-- PICKLETTER "B"          Quit  ?=582  (/ Print;ResetBorders)
-
-- PICKLETTER "F"          Quit  ?=582  (/ Print;ResetDefaults)
-
)
-- PICKLETTER "A"          Stay  ?=582  (/ Print;Align)
-
-- PICKLETTER "G"          Stay  ?=582  (/ Print;Go)
-
-- PICKLETTER "Q"          Quit  ?=582
-
)
-- PICKLETTER "F"          Quit  ?=582  (/ Print;OutputFile)
-
(
-- PICKLETTER "R"          Stay  ?=582  (/ Print;Block)
-
-- PICKLETTER "L"          Stay  ?=582  (/ Print;SkipLine)
-
-- PICKLETTER "P"          Stay  ?=582  (/ Print;FormFeed)
-
-- PICKLETTER "O"          Stay  ?=582
-
(
-- PICKLETTER "H"          Stay  ?=581  (/ Print;Header)
-
-- PICKLETTER "F"          Stay  ?=581  (/ Print;Footer)
-
-- PICKLETTER "M"          Stay  ?=581
-
(
-- PICKLETTER "L"          Quit  ?=581  (/ Print;LeftMargin)
-
-- PICKLETTER "R"          Quit  ?=581  (/ Print;RightMargin)
```

```
-
-- PICKLETTER "T"        Quit  ?=581  (/ Print;TopMargin)
-
-- PICKLETTER "B"        Quit  ?=581  (/ Print;BottomMargin)
-
-                   Quit  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"        Quit  ?=581  (/ Print;Dimensions)
-
)
-- PICKLETTER "B"        Stay  ?=581
-
(
-- PICKLETTER "C"        Quit  ?=581  (/ Print;LeftBorder)
-
-- PICKLETTER "R"        Quit  ?=581  (/ Print;TopBorder)
-
)
-- PICKLETTER "S"        Stay  ?=581  (/ Print;Setup)
-
-- PICKLETTER "P"        Stay  ?=581  (/ Print;PageLength)
-
-- PICKLETTER "O"        Stay  ?=580  (/ Print;Others)
-
-                   Stay  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"        Stay  ?=581  (/ Print;Rotated)
-
-- PICKLETTER "U"        Stay  ?=581  (/ Print;Update)
-
-- PICKLETTER "Q"        Quit  ?=581
-
)
-- PICKLETTER "C"        Stay  ?=582
-
(
-- PICKLETTER "A"        Quit  ?=582  (/ Print;ResetAll)
-
-- PICKLETTER "R"        Quit  ?=582  (/ Print;ResetBlock)
-
-- PICKLETTER "B"        Quit  ?=582  (/ Print;ResetBorders)
-
-- PICKLETTER "F"        Quit  ?=582  (/ Print;ResetDefaults)
-
)
-- PICKLETTER "A"        Stay  ?=582  (/ Print;Align)
-
-- PICKLETTER "G"        Stay  ?=582  (/ Print;Go)
-
-- PICKLETTER "Q"        Quit  ?=582
-
)
-- PICKLETTER "F"        Quit  ?=0
-
```



```
-- PICKLETTER "B"        Quit  ?=582  (/ Print;OutputHOFile)
-
(
-- PICKLETTER "R"        Stay  ?=582  (/ Print;Block)
-
-- PICKLETTER "L"        Stay  ?=582  (/ Print;SkipLine)
-
-- PICKLETTER "P"        Stay  ?=582  (/ Print;FormFeed)
-
-- PICKLETTER "O"        Stay  ?=582
-
(
-- PICKLETTER "H"        Stay  ?=581  (/ Print;Header)
-
-- PICKLETTER "F"        Stay  ?=581  (/ Print;Footer)
-
-- PICKLETTER "M"        Stay  ?=581
-
(
-- PICKLETTER "L"        Quit  ?=581  (/ Print;LeftMargin)
-
-- PICKLETTER "R"        Quit  ?=581  (/ Print;RightMargin)
-
-- PICKLETTER "T"        Quit  ?=581  (/ Print;TopMargin)
-
-- PICKLETTER "B"        Quit  ?=581  (/ Print;BottomMargin)
-
-                   Quit  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"        Quit  ?=581  (/ Print;Dimensions)
-
)
-- PICKLETTER "B"        Stay  ?=581
-
(
-- PICKLETTER "C"        Quit  ?=581  (/ Print;LeftBorder)
-
-- PICKLETTER "R"        Quit  ?=581  (/ Print;TopBorder)
-
)
-- PICKLETTER "S"        Stay  ?=581  (/ Print;Setup)
-
-- PICKLETTER "P"        Stay  ?=581  (/ Print;PageLength)
-
-- PICKLETTER "O"        Stay  ?=580  (/ Print;Others)
-
-                   Stay  ?=0
    DISPLAY-ONLY
-- PICKLETTER "D"        Stay  ?=581  (/ Print;Rotated)
-
-- PICKLETTER "U"        Stay  ?=581  (/ Print;Update)
-
-- PICKLETTER "Q"        Quit  ?=581
-
```

```
)
-- PICKLETTER "C"          Stay  ?=582

(
-- PICKLETTER "A"          Quit  ?=582  (/ Print;ResetAll)
-- PICKLETTER "R"          Quit  ?=582  (/ Print;ResetBlock)
-- PICKLETTER "B"          Quit  ?=582  (/ Print;ResetBorders)
-- PICKLETTER "F"          Quit  ?=582  (/ Print;ResetDefaults)

)
-- PICKLETTER "A"          Stay  ?=582  (/ Print;Align)
-- PICKLETTER "G"          Stay  ?=582  (/ Print;Go)
-- PICKLETTER "Q"          Quit  ?=582

)
-- PICKLETTER "G"          Quit  ?=582  (/ Print;OutputHQ)

(
-- PICKLETTER "R"          Stay  ?=582  (/ Print;Block)
-- PICKLETTER "L"          Stay  ?=582  (/ Print;SkipLine)
-- PICKLETTER "P"          Stay  ?=582  (/ Print;FormFeed)
-- PICKLETTER "O"          Stay  ?=582

(
-- PICKLETTER "H"          Stay  ?=581  (/ Print;Header)
-- PICKLETTER "F"          Stay  ?=581  (/ Print;Footer)
-- PICKLETTER "M"          Stay  ?=581

(
-- PICKLETTER "L"          Quit  ?=581  (/ Print;LeftMargin)
-- PICKLETTER "R"          Quit  ?=581  (/ Print;RightMargin)
-- PICKLETTER "T"          Quit  ?=581  (/ Print;TopMargin)
-- PICKLETTER "B"          Quit  ?=581  (/ Print;BottomMargin)
--                    Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "D"          Quit  ?=581  (/ Print;Dimensions)

)
-- PICKLETTER "B"          Stay  ?=581
```



```
(
-- PICKLETTER "C"          Quit  ?=581  (/ Print;LeftBorder)
-- PICKLETTER "R"          Quit  ?=581  (/ Print;TopBorder)

)
-- PICKLETTER "S"          Stay  ?=581  (/ Print;Setup)
-- PICKLETTER "P"          Stay  ?=581  (/ Print;PageLength)
-- PICKLETTER "O"          Stay  ?=580  (/ Print;Others)
--                    Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "D"          Stay  ?=581  (/ Print;Rotated)
-- PICKLETTER "U"          Stay  ?=581  (/ Print;Update)
-- PICKLETTER "Q"          Quit  ?=581

)
-- PICKLETTER "C"          Stay  ?=582

(
-- PICKLETTER "A"          Quit  ?=582  (/ Print;ResetAll)
-- PICKLETTER "R"          Quit  ?=582  (/ Print;ResetBlock)
-- PICKLETTER "B"          Quit  ?=582  (/ Print;ResetBorders)
-- PICKLETTER "F"          Quit  ?=582  (/ Print;ResetDefaults)

)
-- PICKLETTER "A"          Stay  ?=582  (/ Print;Align)
-- PICKLETTER "G"          Stay  ?=582  (/ Print;Go)
-- PICKLETTER "Q"          Quit  ?=582

)
-- PICKLETTER "S"          Quit  ?=582  (/ Print;OutputPreview)

(
-- PICKLETTER "R"          Stay  ?=582  (/ Print;Block)
-- PICKLETTER "L"          Stay  ?=582  (/ Print;SkipLine)
-- PICKLETTER "P"          Stay  ?=582  (/ Print;FormFeed)
-- PICKLETTER "O"          Stay  ?=582

(
-- PICKLETTER "H"          Stay  ?=581  (/ Print;Header)
```

```
-- PICKLETTER "F"        Stay  ?=581  {/ Print;Footer}
   -
-- PICKLETTER "M"        Stay  ?=581
   -
(
-- PICKLETTER "L"        Quit  ?=581  {/ Print;LeftMargin}
   -
-- PICKLETTER "R"        Quit  ?=581  {/ Print;RightMargin}
   -
-- PICKLETTER "T"        Quit  ?=581  {/ Print;TopMargin}
   -
-- PICKLETTER "B"        Quit  ?=581  {/ Print;BottomMargin}
   -
-                  Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "D"        Quit  ?=581  {/ Print;Dimensions}
   -
)
-- PICKLETTER "B"        Stay  ?=581
   -
(
-- PICKLETTER "C"        Quit  ?=581  {/ Print;LeftBorder}
   -
-- PICKLETTER "R"        Quit  ?=581  {/ Print;TopBorder}
   -
)
-- PICKLETTER "S"        Stay  ?=581  {/ Print;Setup}
   -
-- PICKLETTER "P"        Stay  ?=581  {/ Print;PageLength}
   -
-- PICKLETTER "O"        Stay  ?=580  {/ Print;Others}
   -
-                  Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "D"        Stay  ?=581  {/ Print;Rotated}
   -
-- PICKLETTER "U"        Stay  ?=581  {/ Print;Update}
   -
-- PICKLETTER "Q"        Quit  ?=581
   -
)
-- PICKLETTER "C"        Stay  ?=582
   -
(
-- PICKLETTER "A"        Quit  ?=582  {/ Print;ResetAll}
   -
-- PICKLETTER "R"        Quit  ?=582  {/ Print;ResetBlock}
   -
-- PICKLETTER "B"        Quit  ?=582  {/ Print;ResetBorders}
   -
-- PICKLETTER "F"        Quit  ?=582  {/ Print;ResetDefaults}
   -
)
-- PICKLETTER "A"        Stay  ?=582  {/ Print;Align}
```



```
   -
-- PICKLETTER "G"        Stay  ?=582  {/ Print;Go}
   -
-- PICKLETTER "Q"        Quit  ?=582
   -
)
-- PICKLETTER "C"        Quit  ?=200
   -
(
-- PICKLETTER "D"        Stay  ?=221  {/ GraphPrint;Destination}
   -
(
-- PICKLETTER "F"        Quit  ?=220  {/ GraphPrint;DestIsFile}
   -
-- PICKLETTER "G"        Quit  ?=221  {/ GraphPrint;DestIsPtr}
   -
-- PICKLETTER "S"        Quit  ?=218  {/ GraphPrint;DestIsPreview}
   -
)
-- PICKLETTER "L"        Stay  ?=199
   -
(
-- PICKLETTER "L"        Stay  ?=199  {/ GraphPrint;Left}
   -
-- PICKLETTER "T"        Stay  ?=199  {/ GraphPrint;Top}
   -
-- PICKLETTER "H"        Stay  ?=199  {/ GraphPrint;Height}
   -
-- PICKLETTER "W"        Stay  ?=199  {/ GraphPrint;Width}
   -
-                  Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "D"        Stay  ?=208  {/ GraphPrint;Dimensions}
   -
-- PICKLETTER "O"        Stay  ?=207  {/ GraphPrint;Rotated}
   -
-- PICKLETTER "4"        Stay  ?=209  {/ Hardware;Aspect43}
   -
-                  Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "R"        Stay  ?=199  {/ Print;ResetAll}
   -
-- PICKLETTER "U"        Stay  ?=204  {/ Print;Update}
   -
-- PICKLETTER "Q"        Quit  ?=199
   -
)
-- PICKLETTER "G"        Stay  ?=201  {/ GraphPrint;Go}
   -
-- PICKLETTER "W"        Stay  ?=197
   -
(
-- PICKLETTER "E"        Stay  ?=197  {/ GraphFile;Postscript}
   -
```

```
    PICKLETTER "P"        Stay  ?=197  {/ GraphFile;PIC}
    PICKLETTER "S"        Stay  ?=197  {/ GraphFile;SlideEPS}
    PICKLETTER "X"        Stay  ?=197  {/ GraphFile;PCX}
    PICKLETTER "Q"        Quit  ?=197
  )
  PICKLETTER "N"          Stay  ?=196  {/ GraphPrint;Use}
  PICKLETTER "Q"          Quit  ?=199
)
)
PICKLETTER "G"            Quit  ?=578
(
  PICKLETTER "T"          Stay  ?=297  {/ Graph;Type}
  PICKLETTER "X"          Stay  ?=578  {/ XAxis;Labels}
  PICKLETTER "A"          Stay  ?=578  {/ 1Series;Block}
  PICKLETTER "B"          Stay  ?=578  {/ 2Series;Block}
  PICKLETTER "C"          Stay  ?=578  {/ 3Series;Block}
  PICKLETTER "D"          Stay  ?=578  {/ 4Series;Block}
  PICKLETTER "E"          Stay  ?=578  {/ 5Series;Block}
  PICKLETTER "F"          Stay  ?=578  {/ 6Series;Block}
  PICKLETTER "R"          Stay  ?=578
(
  PICKLETTER "G"          GoUp  ?=578  {/ Graph;ResetAll}
  PICKLETTER "X"          Stay  ?=578  {/ XAxis;Reset}
  PICKLETTER "A"          Stay  ?=578  {/ Graph;Reset1}
  PICKLETTER "B"          Stay  ?=578  {/ Graph;Reset2}
  PICKLETTER "C"          Stay  ?=578  {/ Graph;Reset3}
  PICKLETTER "D"          Stay  ?=578  {/ Graph;Reset4}
  PICKLETTER "E"          Stay  ?=578  {/ Graph;Reset5}
  PICKLETTER "F"          Stay  ?=578  {/ Graph;Reset6}
  PICKLETTER "Q"          Quit  ?=578
```



```
)
  PICKLETTER "V"          Stay  ?=246  {/ Graph;View}
  PICKLETTER "S"          Stay  ?=578  {/ GraphFile;PIC}
  PICKLETTER "O"          Stay  ?=577
(
  PICKLETTER "L"          Stay  ?=286
(
  PICKLETTER "A"          Quit  ?=286  {/ 1Series;Legend}
  PICKLETTER "B"          Quit  ?=286  {/ 2Series;Legend}
  PICKLETTER "C"          Quit  ?=286  {/ 3Series;Legend}
  PICKLETTER "D"          Quit  ?=286  {/ 4Series;Legend}
  PICKLETTER "E"          Quit  ?=286  {/ 5Series;Legend}
  PICKLETTER "F"          Quit  ?=286  {/ 6Series;Legend}
                  Quit  ?=0
    DISPLAY-ONLY
  PICKLETTER "P"          Quit  ?=577  {/ Graph;LegendPos}
  PICKLETTER "T"          Quit  ?=577  {/ GraphPrint;Fonts}
)
  PICKLETTER "F"          Stay  ?=278
(
  PICKLETTER "G"          Stay  ?=278  {/ CompGraph;GraphFormat}
  PICKLETTER "A"          Stay  ?=278  {/ CompGraph;AFormat}
  PICKLETTER "B"          Stay  ?=278  {/ CompGraph;Bformat}
  PICKLETTER "C"          Stay  ?=278  {/ CompGraph;Cformat}
  PICKLETTER "D"          Stay  ?=278  {/ CompGraph;DFormat}
  PICKLETTER "E"          Stay  ?=278  {/ CompGraph;Eformat}
  PICKLETTER "F"          Stay  ?=278  {/ CompGraph;Fformat}
                  Stay  ?=0
    DISPLAY-ONLY
  PICKLETTER "L"          Stay  ?=280
(
  PICKLETTER "A"          Stay  ?=280  {/ 1Series;LineStyle}
```

```
-
-- PICKLETTER "B"          Stay  ?=280  (/ 2Series;LineStyle)
-
-- PICKLETTER "C"          Stay  ?=280  (/ 3Series;LineStyle)
-
-- PICKLETTER "D"          Stay  ?=280  (/ 4Series;LineStyle)
-
-- PICKLETTER "E"          Stay  ?=280  (/ 5Series;LineStyle)
-
-- PICKLETTER "F"          Stay  ?=280  (/ 6Series;LineStyle)
-
-- PICKLETTER "Q"          Quit  ?=280
-
)
-- PICKLETTER "M"          Stay  ?=279
-
(
-- PICKLETTER "A"          Stay  ?=279  (/ 1Series;Markers)
-
-- PICKLETTER "B"          Stay  ?=279  (/ 2Series;Markers)
-
-- PICKLETTER "C"          Stay  ?=279  (/ 3Series;Markers)
-
-- PICKLETTER "D"          Stay  ?=279  (/ 4Series;Markers)
-
-- PICKLETTER "E"          Stay  ?=279  (/ 5Series;Markers)
-
-- PICKLETTER "F"          Stay  ?=279  (/ 6Series;Markers)
-
-- PICKLETTER "Q"          Quit  ?=279
-
)
-- PICKLETTER "Q"          Quit  ?=0
-
)
-- PICKLETTER "T"          Stay  ?=287
-
(
-- PICKLETTER "F"          Quit  ?=287  (/ Graph;MainTitle)
-
-- PICKLETTER "S"          Quit  ?=287  (/ Graph;SubTitle)
-
-- PICKLETTER "X"          Quit  ?=287  (/ XAxis;Title)
-
-- PICKLETTER "Y"          Quit  ?=287  (/ YAxis;Title)
-
-                    Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "2"          Quit  ?=287  (/ Y2Axis;Title)
-
-- PICKLETTER "T"          Quit  ?=285  (/ GraphPrint;Fonts)
-
)
-- PICKLETTER "G"          Stay  ?=255  (/ Graph;GridStatus)
```



```
-
(
-- PICKLETTER "H"          Quit  ?=255  (/ CompGraph;GridHorz)
-
-- PICKLETTER "V"          Quit  ?=255  (/ CompGraph;GridVert)
-
-- PICKLETTER "B"          Quit  ?=255  (/ CompGraph;GridBoth)
-
-- PICKLETTER "C"          Quit  ?=255  (/ CompGraph;GridClear)
-
-                    Quit  ?=0
   DISPLAY-ONLY
-- PICKLETTER "G"          Quit  ?=255  (/ Graph;GridColor)
-
-- PICKLETTER "L"          Quit  ?=255  (/ Graph;GridLines)
-
-- PICKLETTER "F"          Quit  ?=255  (/ Graph;GridFill)
-
)
-- PICKLETTER "S"          Stay  ?=576
-
(
-- PICKLETTER "Y"          Quit  ?=258
-
(
-- PICKLETTER "A"          Stay  ?=264  (/ CompGraph;YAuto)
-
-- PICKLETTER "M"          Stay  ?=264  (/ CompGraph;YManual)
-
-- PICKLETTER "L"          Stay  ?=264  (/ YAxis;Min)
-
-- PICKLETTER "U"          Stay  ?=264  (/ YAxis;Max)
-
-- PICKLETTER "F"          Stay  ?=263  (/ YAxis;Format)
-
-- PICKLETTER "I"          Stay  ?=260  (/ YAxis;ShowScale)
-
-                    Stay  ?=0
   DISPLAY-ONLY
-- PICKLETTER "T"          Stay  ?=264  (/ YAxis;Step)
-
-- PICKLETTER "N"          Stay  ?=262  (/ YAxis;Skip)
-
-- PICKLETTER "S"          Stay  ?=259  (/ YAxis;ScaleType)
-
-- PICKLETTER "Q"          Quit  ?=264
-
)
-- PICKLETTER "X"          Quit  ?=265
-
(
-- PICKLETTER "A"          Stay  ?=264  (/ CompGraph;XAuto)
-
-- PICKLETTER "M"          Stay  ?=264  (/ CompGraph;XManual)
```

```
    -
" PICKLETTER "L"          Stay  ?=264  (/ XAxis;Min)
    -
" PICKLETTER "U"          Stay  ?=264  (/ XAxis;Max)
    -
" PICKLETTER "F"          Stay  ?=263  (/ XAxis;Format)
    -
" PICKLETTER "I"          Stay  ?=260  (/ XAxis;ShowScale)
    -
-                   Stay  ?=0
    DISPLAY-ONLY
" PICKLETTER "T"          Stay  ?=264  (/ XAxis;Step)
    -
" PICKLETTER "S"          Stay  ?=259  (/ XAxis;ScaleType)
    -
" PICKLETTER "D"          Stay  ?=261  (/ XAxis;Alternate)
    -
" PICKLETTER "Q"          Quit  ?=264
    -
)
" PICKLETTER "S"          Quit  ?=262  (/ XAxis;Skip)
    -
-                   Quit  ?=0
    DISPLAY-ONLY
" PICKLETTER "Z"          Quit  ?=257
    -
(
" PICKLETTER "A"          Stay  ?=264  (/ CompGraph;Y2Auto)
    -
" PICKLETTER "M"          Stay  ?=264  (/ CompGraph;Y2Manual)
    -
" PICKLETTER "L"          Stay  ?=264  (/ Y2Axis;Min)
    -
" PICKLETTER "U"          Stay  ?=264  (/ Y2Axis;Max)
    -
" PICKLETTER "F"          Stay  ?=263  (/ Y2Axis;Format)
    -
" PICKLETTER "I"          Stay  ?=260  (/ Y2Axis;ShowScale)
    -
-                   Stay  ?=0
    DISPLAY-ONLY
" PICKLETTER "T"          Stay  ?=264  (/ Y2Axis;Step)
    -
" PICKLETTER "N"          Stay  ?=262  (/ Y2Axis;Skip)
    -
" PICKLETTER "S"          Stay  ?=259  (/ Y2Axis;ScaleType)
    -
" PICKLETTER "Q"          Quit  ?=264
    -
)
)
" PICKLETTER "C"          Stay  ?=577  (/ Graph;Color)
    -
" PICKLETTER "B"          Stay  ?=577  (/ Graph;BW)
```



```
    -
" PICKLETTER "D"          Stay  ?=577
    -
(
" PICKLETTER "A"          Stay  ?=276  (/ CompGraph;ALabels)
    -
" PICKLETTER "B"          Stay  ?=276  (/ CompGraph;BLabels)
    -
" PICKLETTER "C"          Stay  ?=276  (/ CompGraph;CLabels)
    -
" PICKLETTER "D"          Stay  ?=276  (/ CompGraph;DLabels)
    -
" PICKLETTER "E"          Stay  ?=276  (/ CompGraph;ELabels)
    -
" PICKLETTER "F"          Stay  ?=276  (/ CompGraph;FLabels)
    -
-                   Stay  ?=0
    DISPLAY-ONLY
" PICKLETTER "T"          Stay  ?=285  (/ GraphPrint;Fonts)
    -
" PICKLETTER "Q"          Quit  ?=577
    -
)
-                   Stay  ?=0
    DISPLAY-ONLY
" PICKLETTER "R"          Stay  ?=284
    -
(
" PICKLETTER "C"          Stay  ?=283
    -
(
" PICKLETTER "A"          Stay  ?=283  (/ 1Series;Color)
    -
" PICKLETTER "B"          Stay  ?=283  (/ 2Series;Color)
    -
" PICKLETTER "C"          Stay  ?=283  (/ 3Series;Color)
    -
" PICKLETTER "D"          Stay  ?=283  (/ 4Series;Color)
    -
" PICKLETTER "E"          Stay  ?=283  (/ 5Series;Color)
    -
" PICKLETTER "F"          Stay  ?=283  (/ 6Series;Color)
    -
" PICKLETTER "Q"          Quit  ?=284
    -
)
" PICKLETTER "F"          Stay  ?=282
    -
(
" PICKLETTER "A"          Stay  ?=282  (/ 1Series;Pattern)
    -
" PICKLETTER "B"          Stay  ?=282  (/ 2Series;Pattern)
    -
" PICKLETTER "C"          Stay  ?=282  (/ 3Series;Pattern)
```

```
PICKLETTER "D"      Stay  ?=282  (/ 4Series;Pattern)
PICKLETTER "E"      Stay  ?=282  (/ 5Series;Pattern)
PICKLETTER "F"      Stay  ?=282  (/ 6Series;Pattern)
PICKLETTER "Q"      Quit  ?=284
)
PICKLETTER "O"      Stay  ?=278
(
PICKLETTER "A"      Stay  ?=278  (/ 1Series;Type)
PICKLETTER "B"      Stay  ?=278  (/ 2Series;Type)
PICKLETTER "C"      Stay  ?=278  (/ 3Series;Type)
PICKLETTER "D"      Stay  ?=278  (/ 4Series;Type)
PICKLETTER "E"      Stay  ?=278  (/ 5Series;Type)
PICKLETTER "F"      Stay  ?=278  (/ 6Series;Type)
PICKLETTER "Q"      Quit  ?=284
)
PICKLETTER "Y"      Stay  ?=284
(
PICKLETTER "A"      Stay  ?=274  (/ 1Series;YAxis)
PICKLETTER "B"      Stay  ?=274  (/ 2Series;YAxis)
PICKLETTER "C"      Stay  ?=274  (/ 3Series;YAxis)
PICKLETTER "D"      Stay  ?=274  (/ 4Series;YAxis)
PICKLETTER "E"      Stay  ?=274  (/ 5Series;YAxis)
PICKLETTER "F"      Stay  ?=274  (/ 6Series;YAxis)
PICKLETTER "Q"      Quit  ?=284
)
PICKLETTER "P"      Stay  ?=273
(
PICKLETTER "L"      Quit  ?=272  (/ Pie;ValueFormat)
PICKLETTER "E"      Quit  ?=271
(
```



```
PICKLETTER "1"      Stay  ?=271  (/ PieExploded;1)
PICKLETTER "2"      Stay  ?=271  (/ PieExploded;2)
PICKLETTER "3"      Stay  ?=271  (/ PieExploded;3)
PICKLETTER "4"      Stay  ?=271  (/ PieExploded;4)
PICKLETTER "5"      Stay  ?=271  (/ PieExploded;5)
PICKLETTER "6"      Stay  ?=271  (/ PieExploded;6)
PICKLETTER "7"      Stay  ?=271  (/ PieExploded;7)
PICKLETTER "8"      Stay  ?=271  (/ PieExploded;8)
PICKLETTER "9"      Stay  ?=271  (/ PieExploded;9)
PICKLETTER "Q"      Quit  ?=273
)
PICKLETTER "P"      Quit  ?=270
(
PICKLETTER "1"      Stay  ?=270  (/ PiePattern;1)
PICKLETTER "2"      Stay  ?=270  (/ PiePattern;2)
PICKLETTER "3"      Stay  ?=270  (/ PiePattern;3)
PICKLETTER "4"      Stay  ?=270  (/ PiePattern;4)
PICKLETTER "5"      Stay  ?=270  (/ PiePattern;5)
PICKLETTER "6"      Stay  ?=270  (/ PiePattern;6)
PICKLETTER "7"      Stay  ?=270  (/ PiePattern;7)
PICKLETTER "8"      Stay  ?=270  (/ PiePattern;8)
PICKLETTER "9"      Stay  ?=270  (/ PiePattern;9)
PICKLETTER "Q"      Quit  ?=273
)
PICKLETTER "C"      Quit  ?=269
(
PICKLETTER "1"      Stay  ?=269  (/ PieColor;1)
PICKLETTER "2"      Stay  ?=269  (/ PieColor;2)
PICKLETTER "3"      Stay  ?=269  (/ PieColor;3)
```

```
- PICKLETTER "4"        Stay  ?=269  (/ PieColor;4)
- PICKLETTER "5"        Stay  ?=269  (/ PieColor;5)
- PICKLETTER "6"        Stay  ?=269  (/ PieColor;6)
- PICKLETTER "7"        Stay  ?=269  (/ PieColor;7)
- PICKLETTER "8"        Stay  ?=269  (/ PieColor;8)
- PICKLETTER "9"        Stay  ?=269  (/ PieColor;9)
- PICKLETTER "Q"        Quit  ?=273
)
- PICKLETTER "T"        Quit  ?=268  (/ Pie;TickMarks)
)
- PICKLETTER "B"        Stay  ?=277  (/ Graph;BarWidth)
- PICKLETTER "Q"        Quit  ?=284
)
- PICKLETTER "O"        Stay  ?=256
(
- PICKLETTER "O"        Quit  ?=254
(
- PICKLETTER "T"        Quit  ?=254  (/ Graph;TitleOtl)
- PICKLETTER "L"        Quit  ?=254  (/ Graph;LegendOtl)
- PICKLETTER "G"        Quit  ?=254  (/ Graph;GraphOtl)
)
- PICKLETTER "B"        Quit  ?=253  (/ Graph;BackColor)
- PICKLETTER "T"        Quit  ?=252  (/ Graph;3D)
)
- PICKLETTER "A"        Stay  ?=244  (/ Graph;Annotate)
- PICKLETTER "I"        Stay  ?=251  (/ Graph;NameInsert)
- PICKLETTER "H"        Stay  ?=250  (/ Graph;NameHide)
- PICKLETTER "U"        Stay  ?=267  (/ Graph;UpdateGraph)
- PICKLETTER "Q"        Quit  ?=284
)
- PICKLETTER "N"        Stay  ?=249
```



```
(
- PICKLETTER "U"        Quit  ?=249  (/ Graph;NameUse)
- PICKLETTER "C"        Quit  ?=249  (/ Graph;NameCreate)
- PICKLETTER "D"        Quit  ?=249  (/ Graph;NameDelete)
- PICKLETTER "S"        Quit  ?=249  (/ Graph;NameSide)
- PICKLETTER "R"        Quit  ?=249  (/ Graph;NameReset)
- PICKLETTER "G"        Quit  ?=249  (/ Graph;NameCopy)
)
-                       Stay  ?=0
  DISPLAY-ONLY
- PICKLETTER "I"        Stay  ?=245  (/ Graph;FastGraph)
- PICKLETTER "Q"        Quit  ?=284
)
- PICKLETTER "D"        Quit  ?=575
(
- PICKLETTER "F"        Quit  ?=575  (/ Math;Fill)
- PICKLETTER "I"        Quit  ?=575
(
- PICKLETTER "1"        Quit  ?=109  (/ Math;1CellWhat-If)
- PICKLETTER "2"        Quit  ?=108  (/ Math;2CellWhat-If)
- PICKLETTER "R"        Quit  ?=110  (/ Math;ResetWhat-If)
)
- PICKLETTER "S"        Quit  ?=575
(
- PICKLETTER "D"        Stay  ?=194  (/ Sort;Block)
- PICKLETTER "P"        Stay  ?=194  (/ Sort;Key1)
- PICKLETTER "S"        Stay  ?=194  (/ Sort;Key2)
- PICKLETTER "3"        Stay  ?=194  (/ Sort;Key3)
- PICKLETTER "4"        Stay  ?=194  (/ Sort;Key4)
- PICKLETTER "5"        Stay  ?=194  (/ Sort;Key5)
- PICKLETTER "R"        Stay  ?=194  (/ Sort;Reset)
- PICKLETTER "G"        Stay  ?=194  (/ Sort;Go)
```

```
-
-                    Stay  ?=0
    DISPLAY-ONLY
- PICKLETTER "O"          Stay  ?=194
-
(
- PICKLETTER "N"          Stay  ?=191  (/ Startup;CellOrder)
-
- PICKLETTER "L"          Stay  ?=191  (/ Startup;LabelOrder)
-
- PICKLETTER "Q"          Quit  ?=194
-
)
- PICKLETTER "Q"          Quit  ?=194
-
)
- PICKLETTER "Q"          Quit  ?=575
-
(
- PICKLETTER "I"          Stay  ?=188  (/ Query;Block)
-
- PICKLETTER "C"          Stay  ?=186  (/ Query;CriteriaBlock)
-
- PICKLETTER "O"          Stay  ?=183  (/ Query;Output)
-
- PICKLETTER "A"          Stay  ?=182  (/ Query;AssignNames)
-
- PICKLETTER "F"          Stay  ?=181  (/ Query;Locate)
-
- PICKLETTER "E"          Stay  ?=180  (/ Query;Extract)
-
- PICKLETTER "U"          Stay  ?=180  (/ Query;Unique)
-
- PICKLETTER "D"          Stay  ?=179  (/ Query;Delete)
-
- PICKLETTER "R"          Stay  ?=178  (/ Query;Reset)
-
- PICKLETTER "Q"          Quit  ?=190
-
)
- PICKLETTER "D"          Quit  ?=575  (/ Math;Distribution)
-
- PICKLETTER "M"          Quit  ?=575
-
(
- PICKLETTER "I"          Quit  ?=123  (/ Math;InvertMatrix)
-
- PICKLETTER "M"          Quit  ?=122  (/ Math;MultiplyMatrix)
-
)
- PICKLETTER "R"          Quit  ?=575
-
(
- PICKLETTER "X"          Stay  ?=126  (/ Regression;Independent)
```



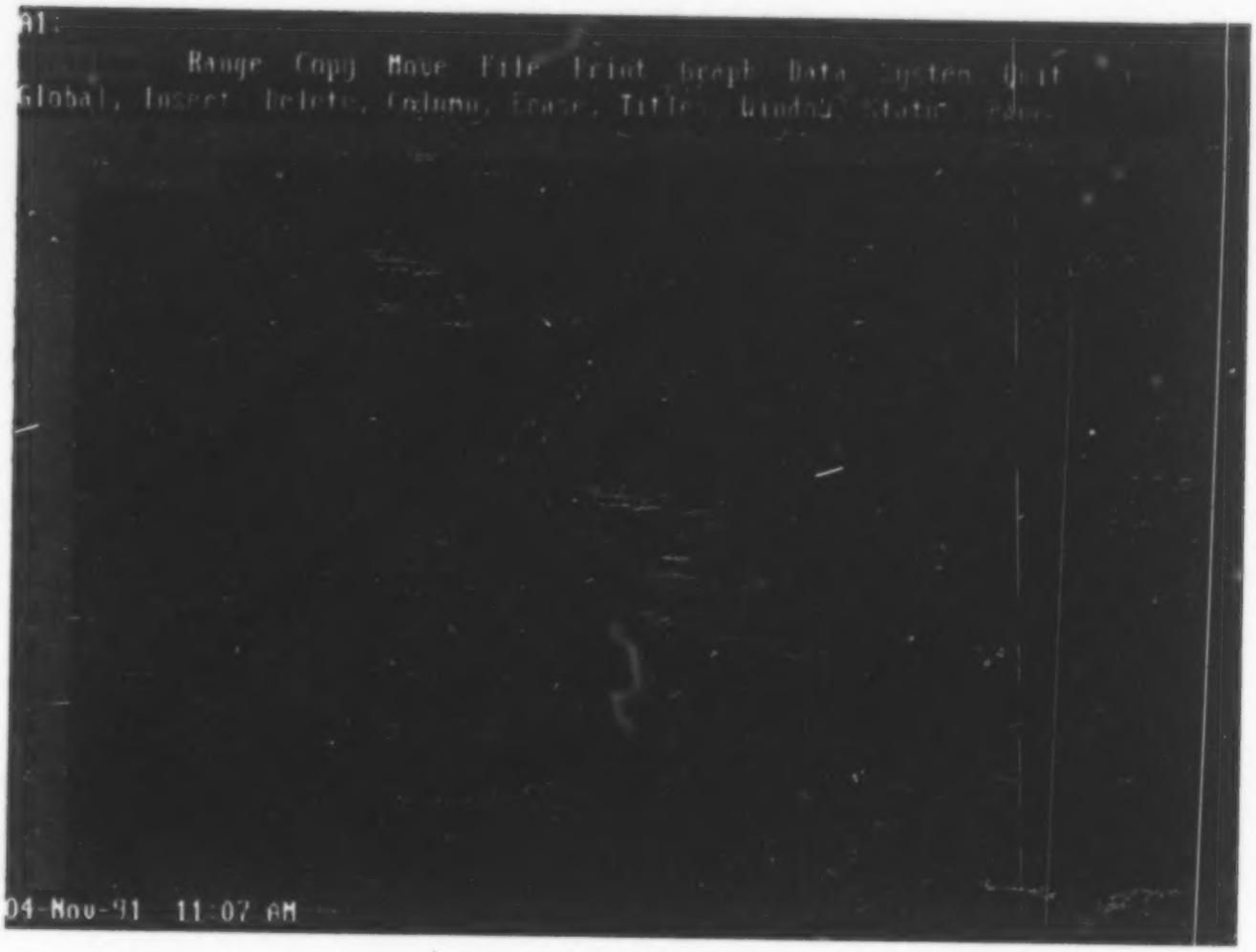
```
-
- PICKLETTER "Y"          Stay  ?=126  (/ Regression;Dependent)
-
- PICKLETTER "O"          Stay  ?=126  (/ Regression;Output)
-
- PICKLETTER "I"          Stay  ?=125  (/ Regression;Intercept)
-
- PICKLETTER "R"          Stay  ?=127  (/ Regression;Reset)
-
- PICKLETTER "G"          Stay  ?=127  (/ Regression;Go)
-
- PICKLETTER "Q"          Quit  ?=127
-
)
- PICKLETTER "P"          Quit  ?=575
-
(
- PICKLETTER "F"          Stay  ?=116
-
(
- PICKLETTER "C"          Quit  ?=114  (/ Parse;CreateLine)
-
- PICKLETTER "E"          Quit  ?=112  (/ Parse;EditLine)
-
)
- PICKLETTER "I"          Stay  ?=115  (/ Parse;Input)
-
- PICKLETTER "O"          Stay  ?=115  (/ Parse;Output)
-
- PICKLETTER "R"          Stay  ?=116  (/ Parse;Reset)
-
- PICKLETTER "G"          Stay  ?=111  (/ Parse;Go)
-
- PICKLETTER "Q"          Quit  ?=116
-
)
-                    Quit  ?=0
    DISPLAY-ONLY
- PICKLETTER "O"          Quit  ?=0
-
(
- PICKLETTER "L"          Stay  ?=120  (/ Optimization;Coefficients)
-
- PICKLETTER "I"          Stay  ?=120  (/ Optimization;Relations)
-
- PICKLETTER "C"          Stay  ?=120  (/ Optimization;Constants)
-
- PICKLETTER "B"          Stay  ?=119  (/ Optimization;Bounds)
-
- PICKLETTER "F"          Stay  ?=119  (/ Optimization;Formulas)
-
- PICKLETTER "O"          Stay  ?=119  (/ Optimization;Objective)
-
- PICKLETTER "E"          Stay  ?=119  (/ Optimization;Extremum)
```

```
                Stay  ?=0
    DISPLAY-ONLY
"" PICKLETTER "S"       Stay  ?=118  {/ Optimization;Solution}
"" PICKLETTER "V"       Stay  ?=118  {/ Optimization;Variables}
"" PICKLETTER "D"       Stay  ?=118  {/ Optimization;Dual}
"" PICKLETTER "A"       Stay  ?=118  {/ Optimization;Additional}
                Stay  ?=0
    DISPLAY-ONLY
"" PICKLETTER "G"       Stay  ?=117  {/ Optimization;Go}
"" PICKLETTER "R"       Stay  ?=117  {/ Optimization;Reset}
"" PICKLETTER "Q"       Quit  ?=117
)
"" PICKLETTER "V"       Quit  ?=105
(
"" PICKLETTER "F"       Stay  ?=104  {/ Math;SolveFormula}
"" PICKLETTER "T"       Stay  ?=104  {/ Math;SolveTarget}
"" PICKLETTER "V"       Stay  ?=104  {/ Math;SolveVariable}
"" PICKLETTER "P"       Stay  ?=104
(
"" PICKLETTER "M"       Quit  ?=104  {/ Math;SolveMaxIt}
"" PICKLETTER "A"       Quit  ?=104  {/ Math;SolveAccuracy}
)
"" PICKLETTER "G"       Stay  ?=104  {/ Math;SolveGo}
"" PICKLETTER "R"       Stay  ?=104  {/ Math;SolveReset}
"" PICKLETTER "Q"       Quit  ?=104
)
"" PICKLETTER "A"       Quit  ?=175
(
"" PICKLETTER "G"       Stay  ?=173  {/ Paradox;SwitchGo}
"" PICKLETTER "L"       Stay  ?=172  {/ Paradox;SwitchFile}
"" PICKLETTER "A"       Stay  ?=171  {/ Paradox;SwitchAutoLoad}
"" PICKLETTER "Q"       Quit  ?=169
```

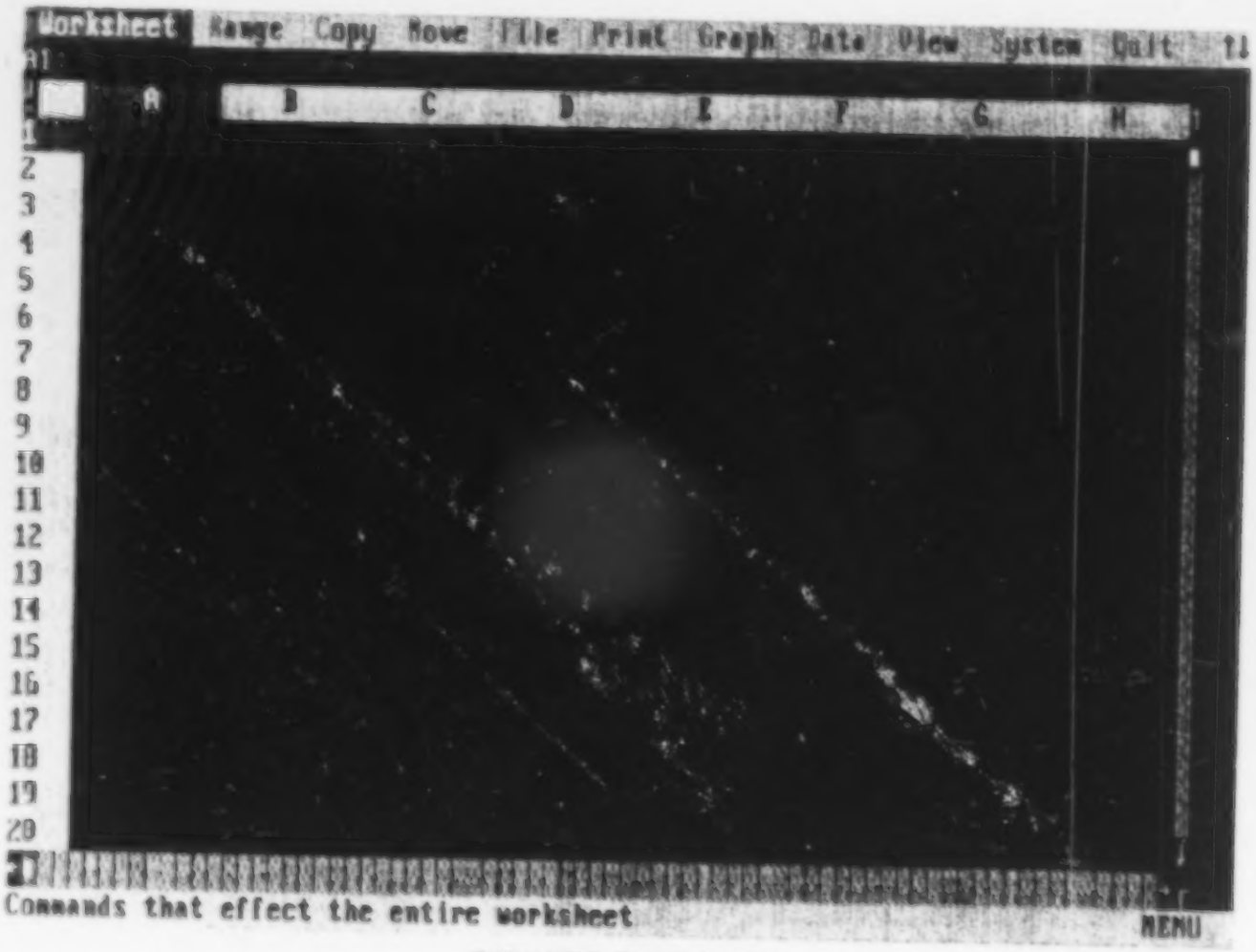


```
)
)
"" PICKLETTER "V"       Quit  ?=574
(
"" PICKLETTER "Z"       Quit  ?=574  {/ View;Zoom}
"" PICKLETTER "T"       Quit  ?=574  {/ View;Arrange}
"" PICKLETTER "S"       Quit  ?=574  {/ View;Cascade}
"" PICKLETTER "M"       Quit  ?=574  {/ View;Size}
"" PICKLETTER "P"       Quit  ?=574  {/ View;Choose}
                Quit  ?=574
    DISPLAY-ONLY
"" PICKLETTER "C"       Quit  ?=574  {/ Basics;Close}
"" PICKLETTER "L"       Quit  ?=574  {/ System;TidyUp}
"" PICKLETTER "X"       Quit  ?=574  {/ System;Exit}
)
"" PICKLETTER "S"       Quit  ?=573
(
"" PICKLETTER "O"       Quit  ?=573  {/ Basics;OS}
                Quit  ?=0
    DISPLAY-ONLY
"" PICKLETTER "D"       Quit  ?=83   {/ ScreenHardware;TextScreenMode}
"" PICKLETTER "F"       Quit  ?=370  {/ View;NewFileMgr}
)
"" PICKLETTER "Q"       Quit  ?=593  {/ Basics;Quit}
)
)
```

Exhibits from Brief of Plaintiff/Appellee from Court of Appeals for the First Circuit

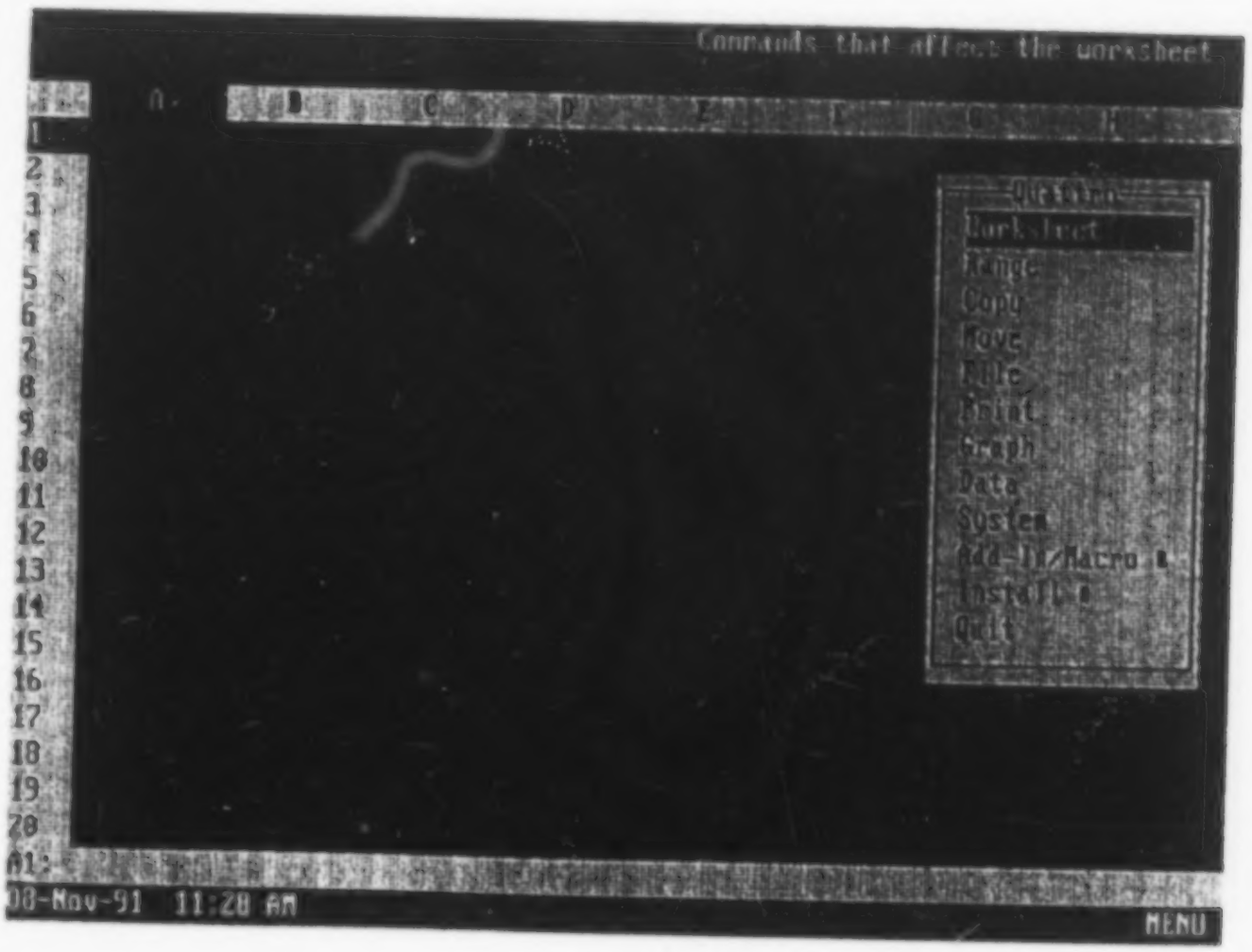

A1:
Range Copy Move File Print Graph Data System Quit
Global, Insert, Delete, Columns, Erase, Title, Window, Status, Page
04-Nov-91 11:07 AM

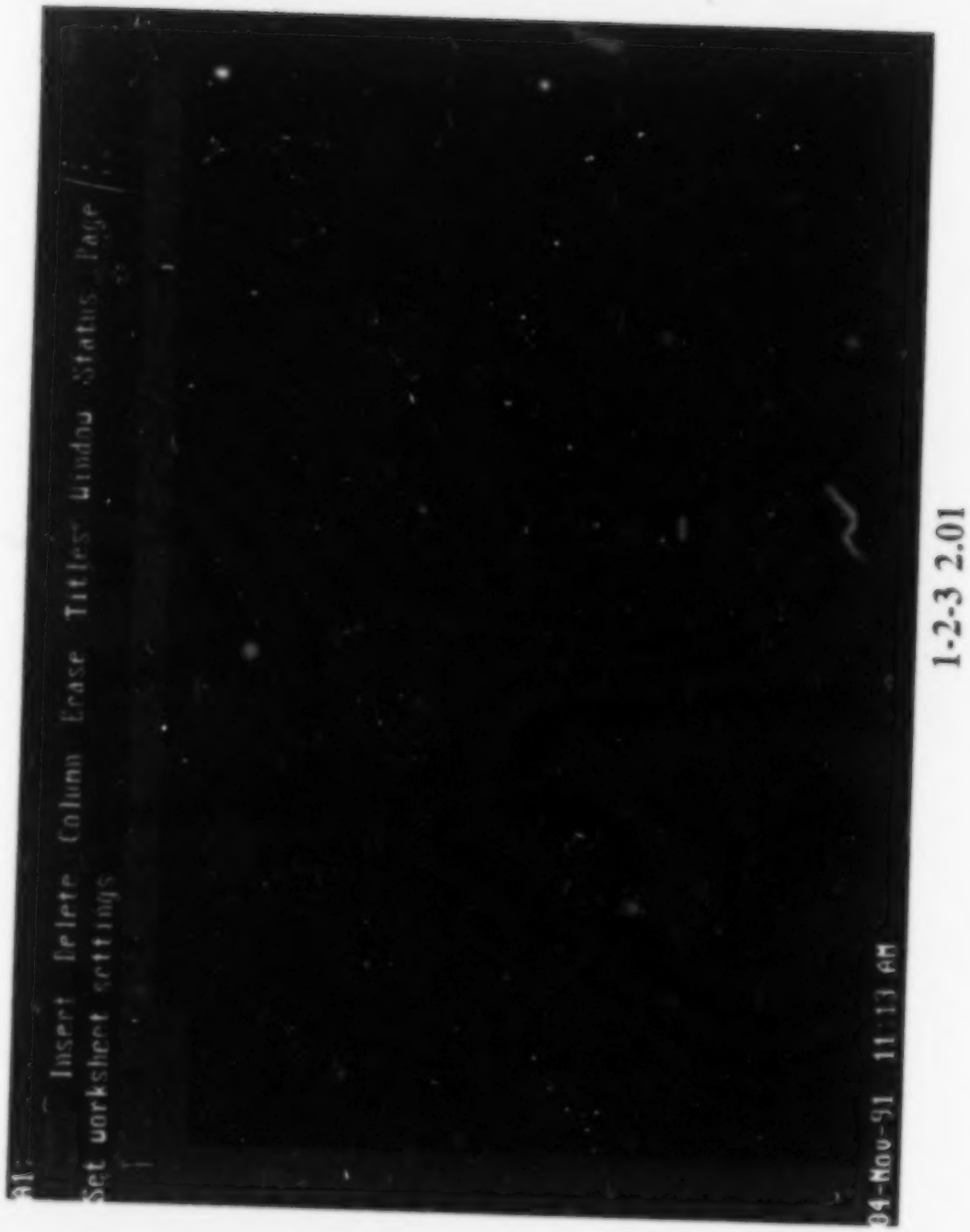
1-2-3 2.01



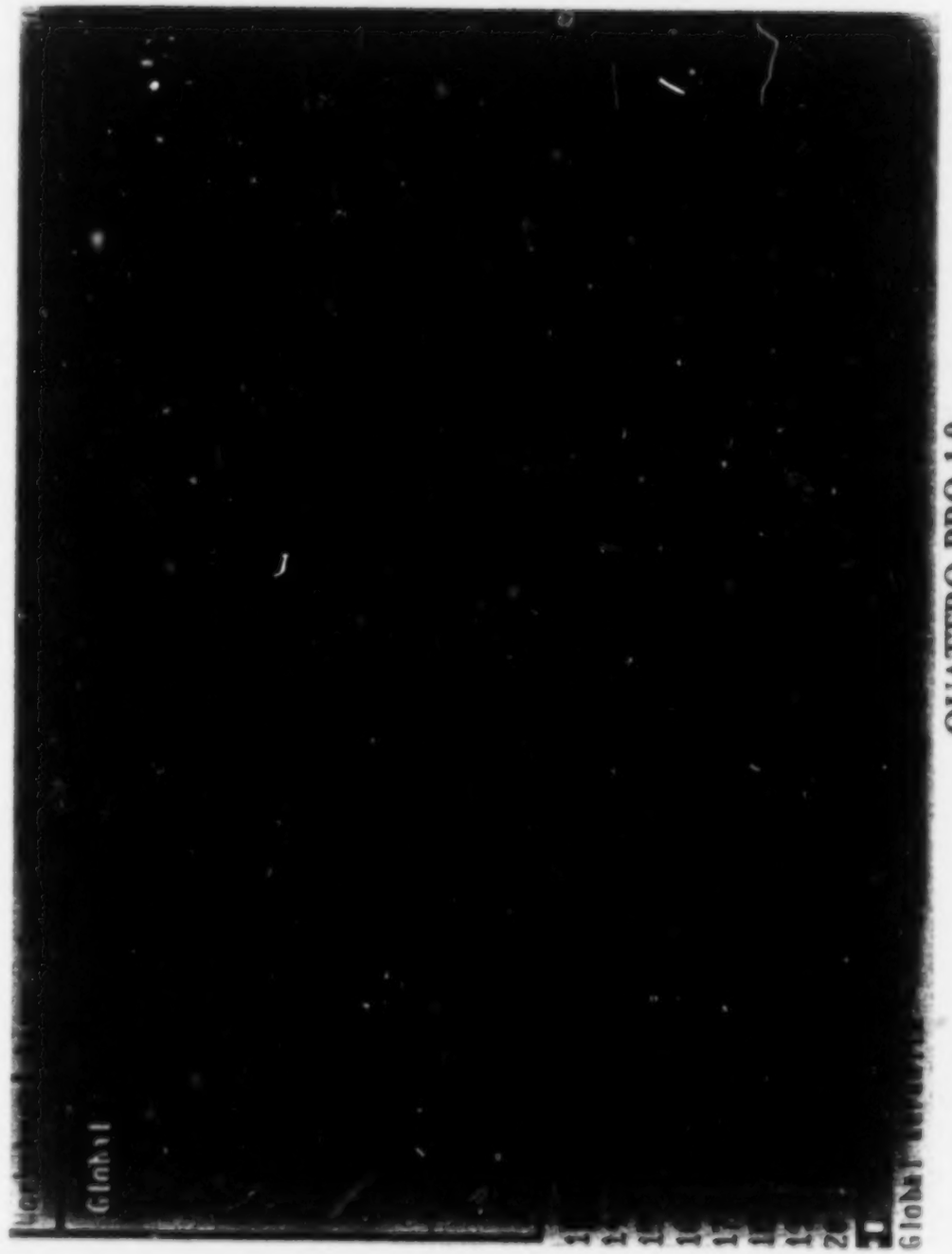
QUATTRO PRO 1.0

QUATTRO

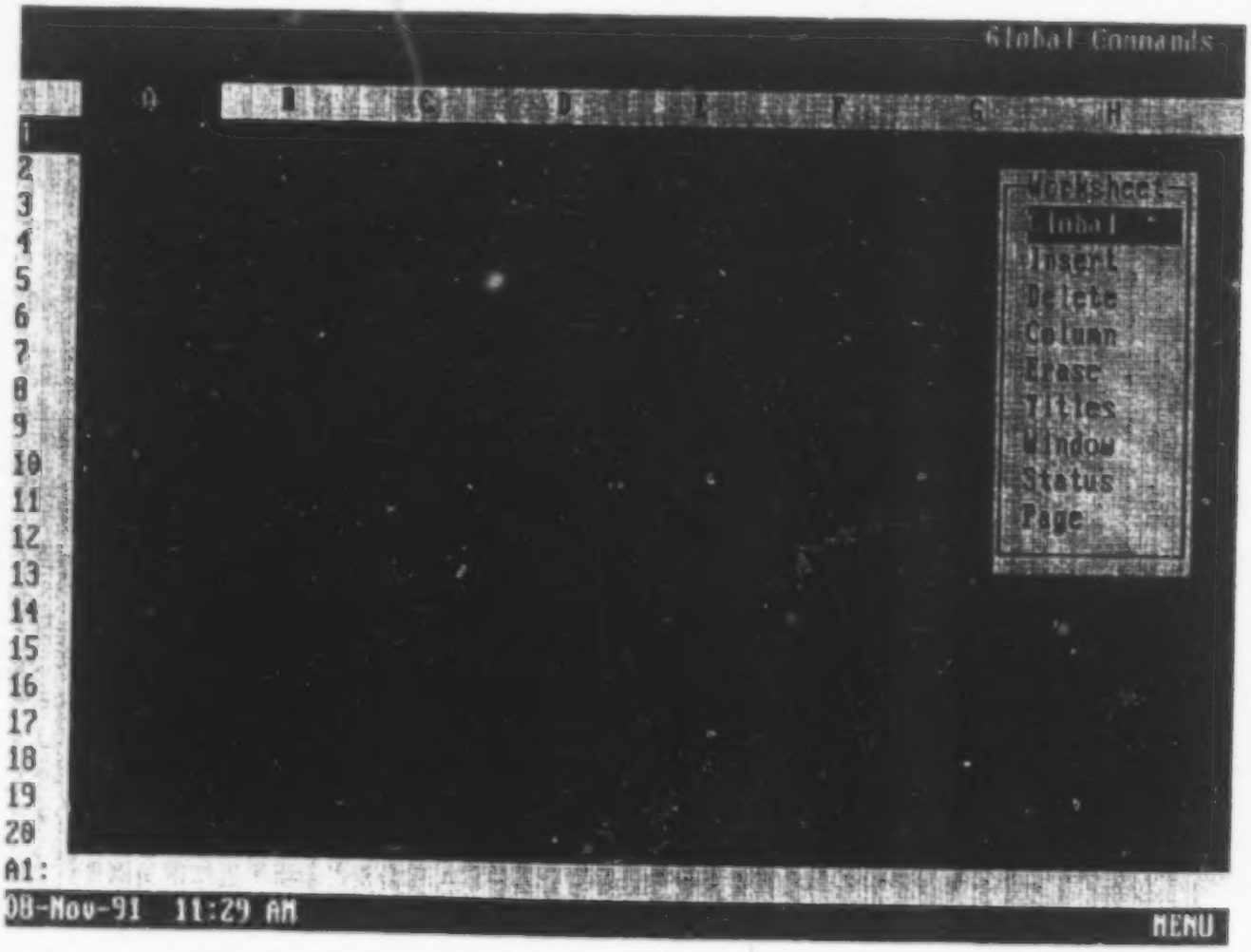

Commands that affect the worksheet
Worksheet
Range
Copy
Move
File
Print
Graph
Data
System
Add-In/Macro
Install
Quit
08-Nov-91 11:20 AM
MENU


Titles Window Status Page
Insert Delete Column Erase Titl
Set worksheet settings
04-Nov-91 11:13 AM
1-2-3 2.01

J.A. 973

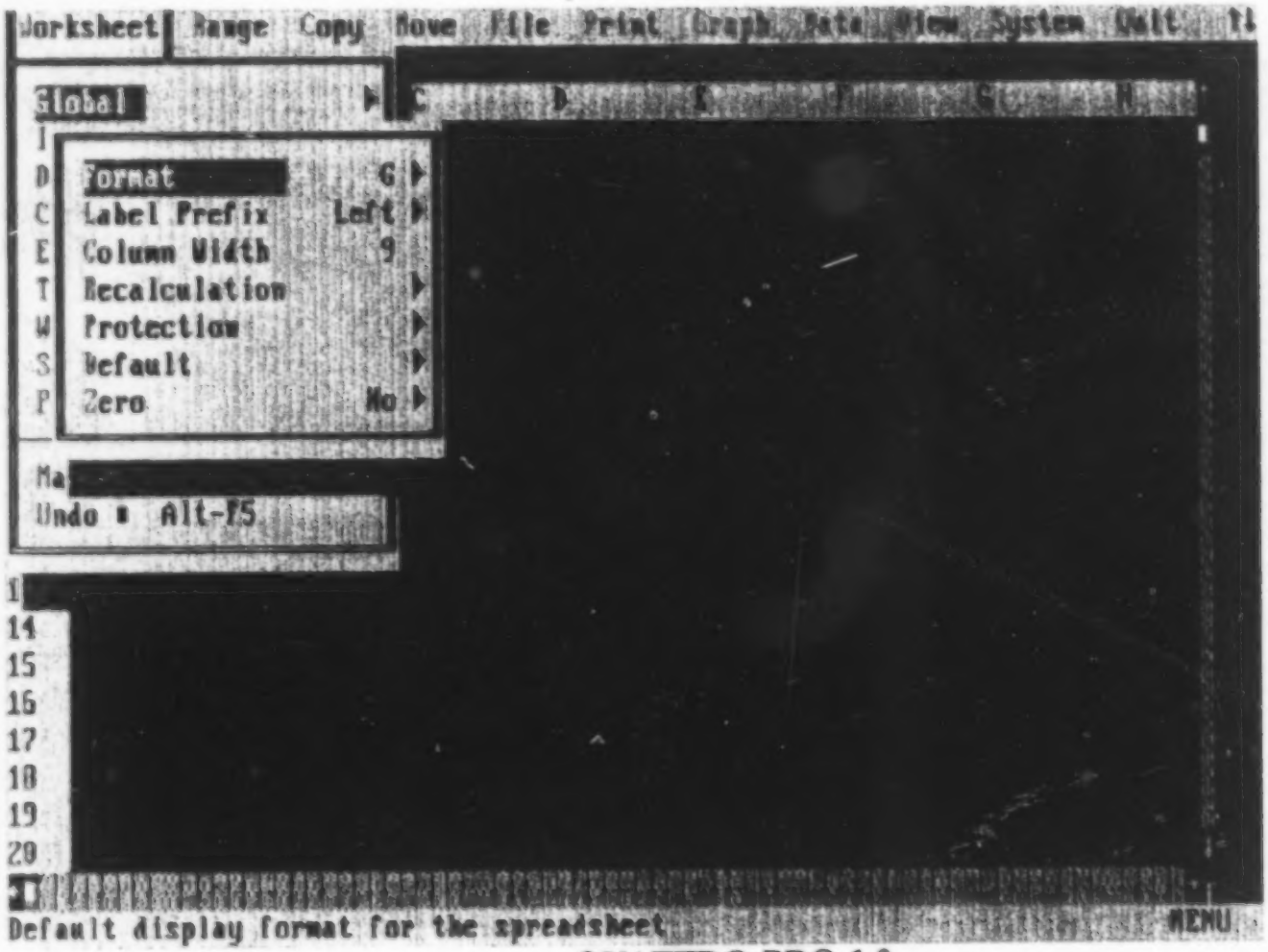


QUATTRO



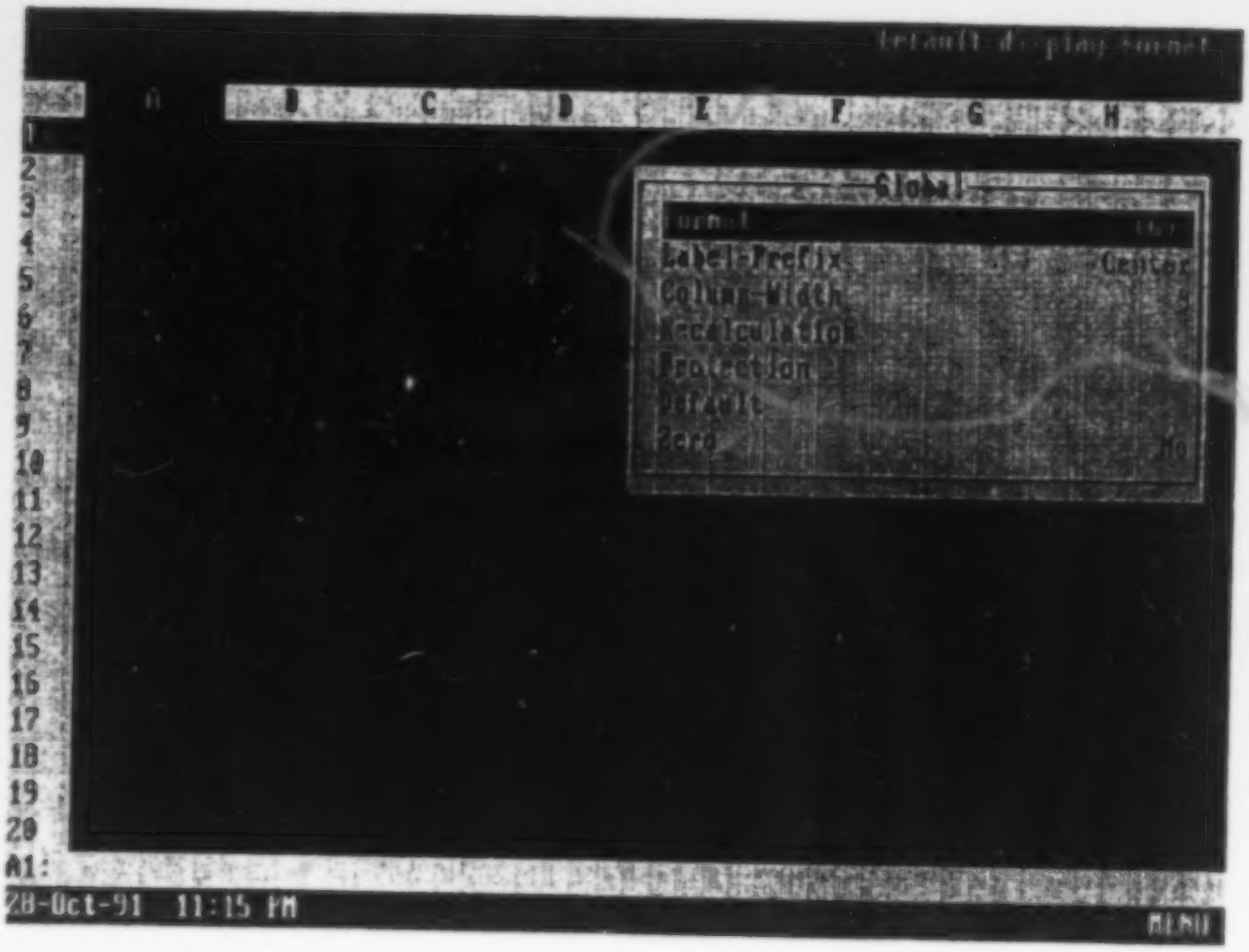
J.A. 974


A1:
Label-Prefix Column-Width Recalculation Protection Default Zero
Set global format
04-Nov-91 02:02 PM



QUATTRO PRO 1.0

QUATTRO



J.A. 977

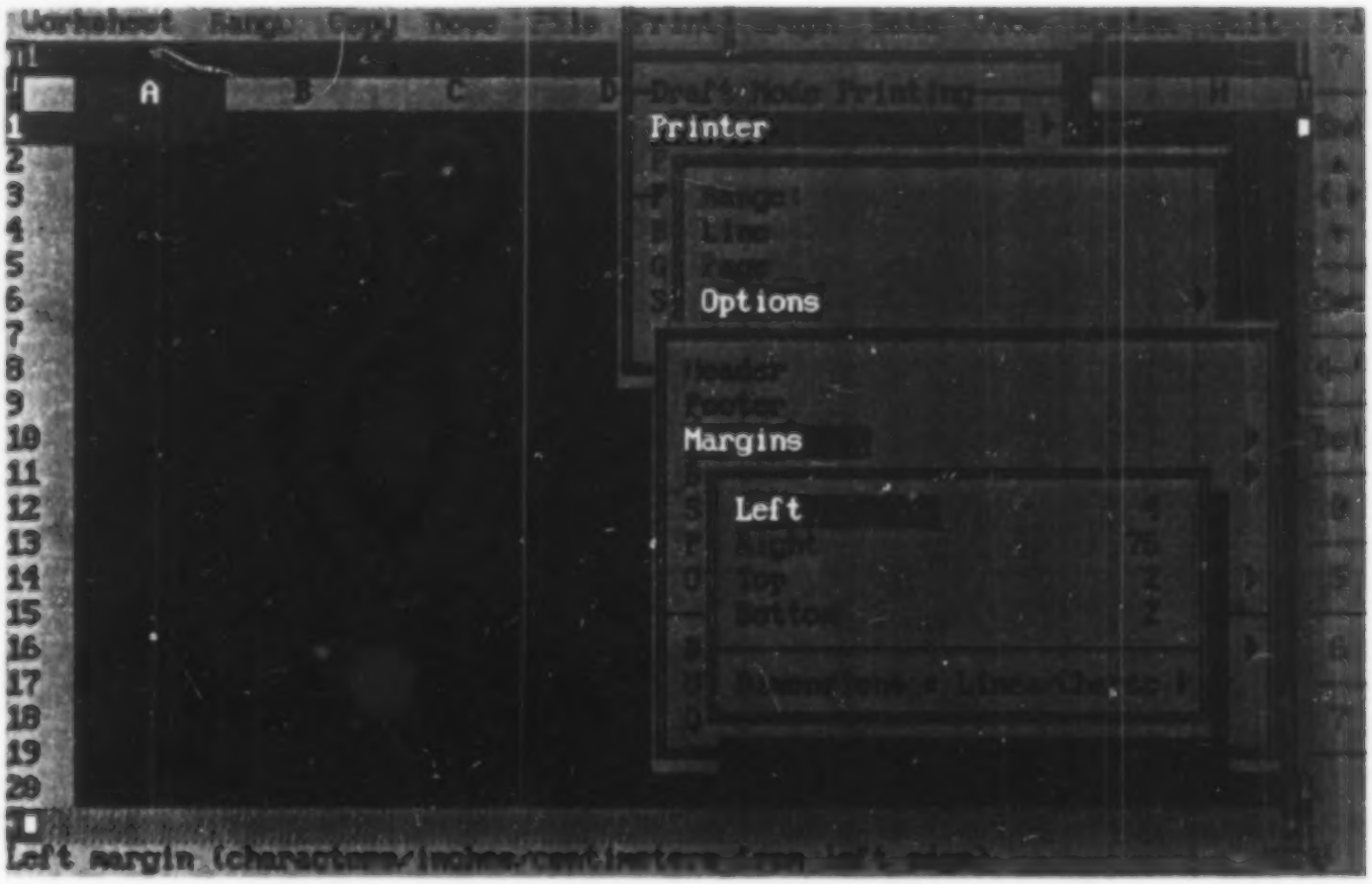

A1
help edit name abs goto window data table recalculate graph
A
1
3 Range
4 Copy
5 Move
6 File
7 Print
8 Graph
9 Data
0 Tools
System
Quit
2 Global
3 Insert
4 Delete
5 Col-Width
6 Erase
7 Titles
8 Window
9 Status
0 Hotkey
Page
218K
14:05

VP PLANNER +

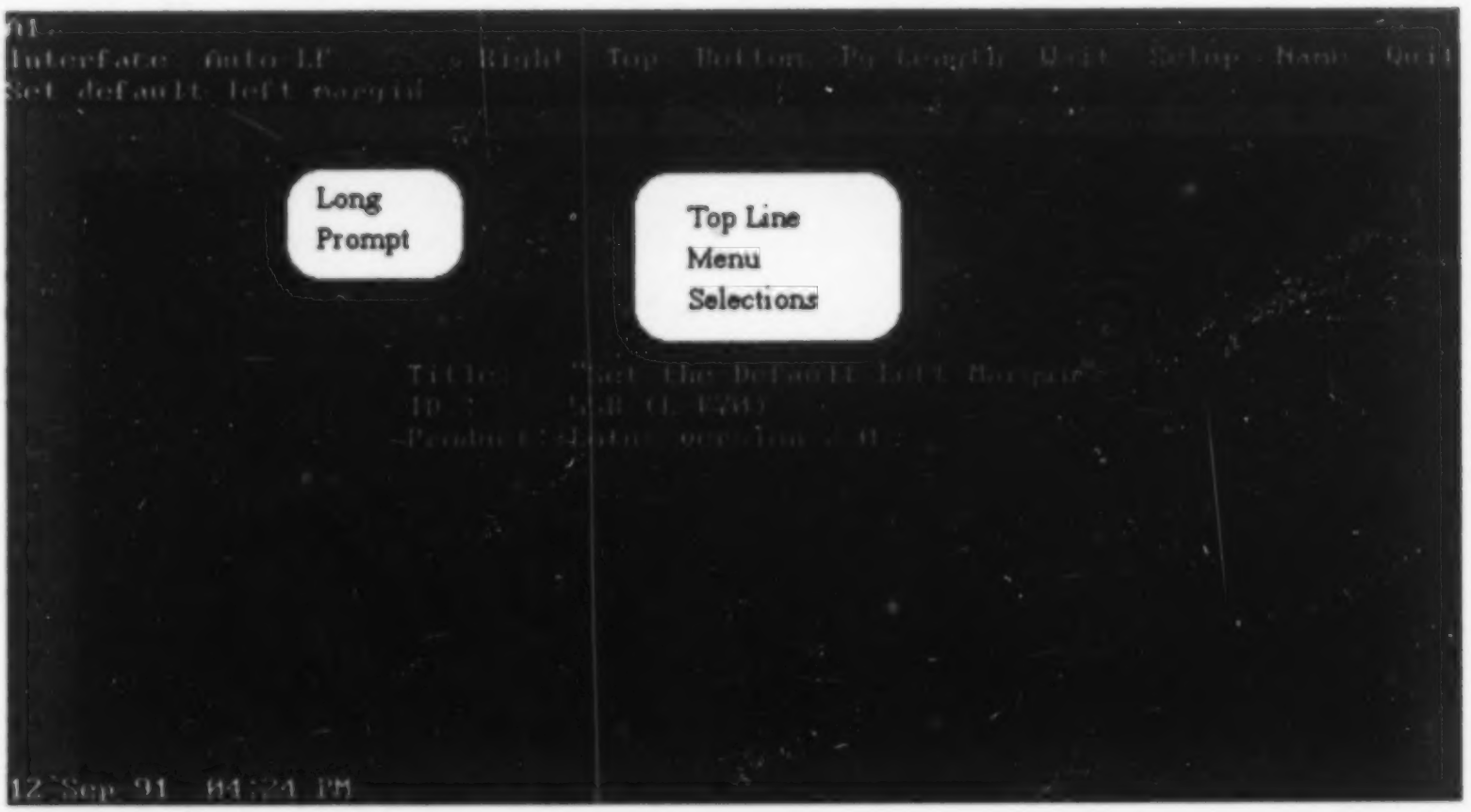
Exhibits from Briefs of Defendant/Appellant from Court of Appeals for the First Circuit



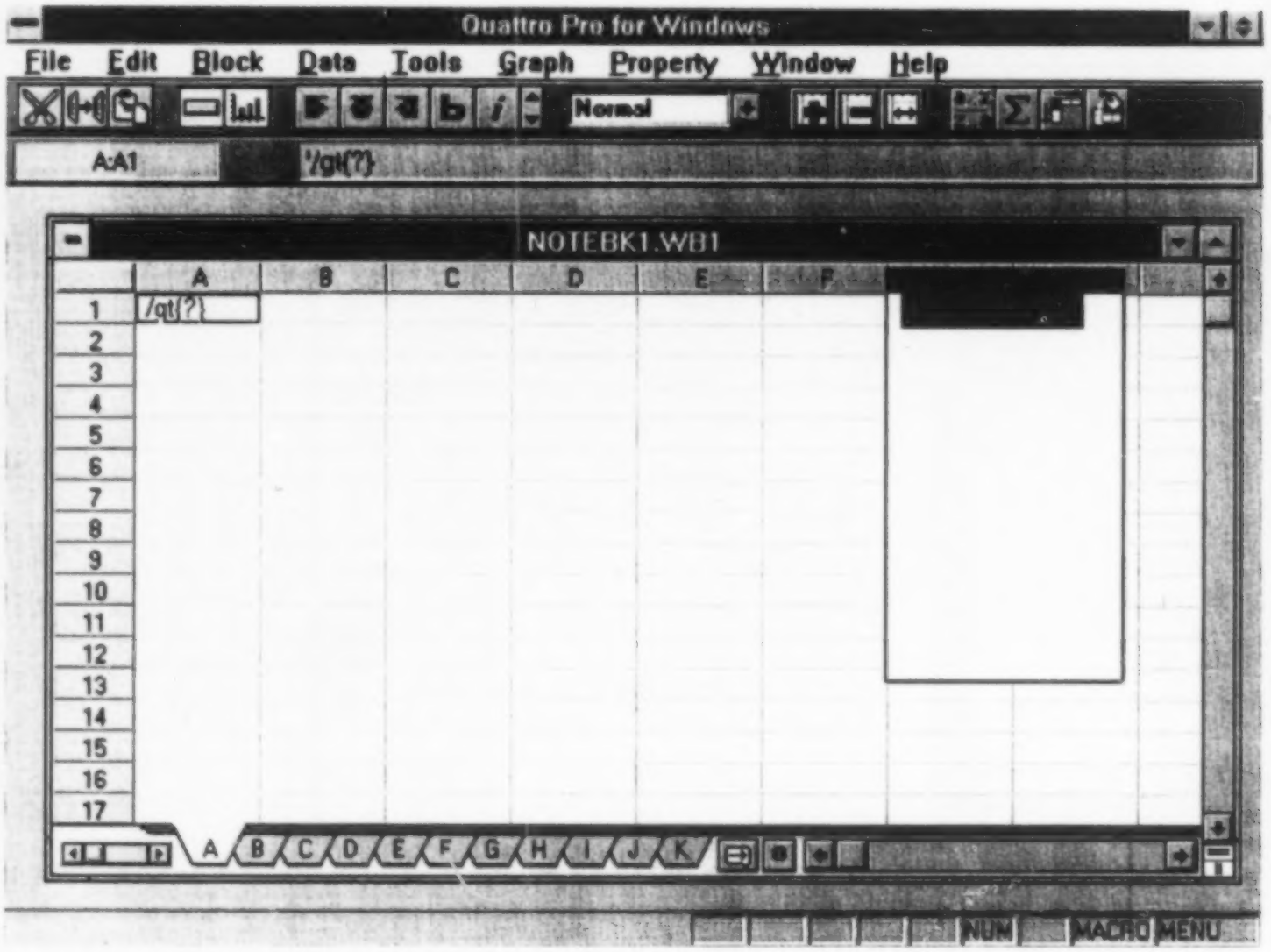
Lotus 1-2-3 Screen Display When Setting the Left Margin for Printing



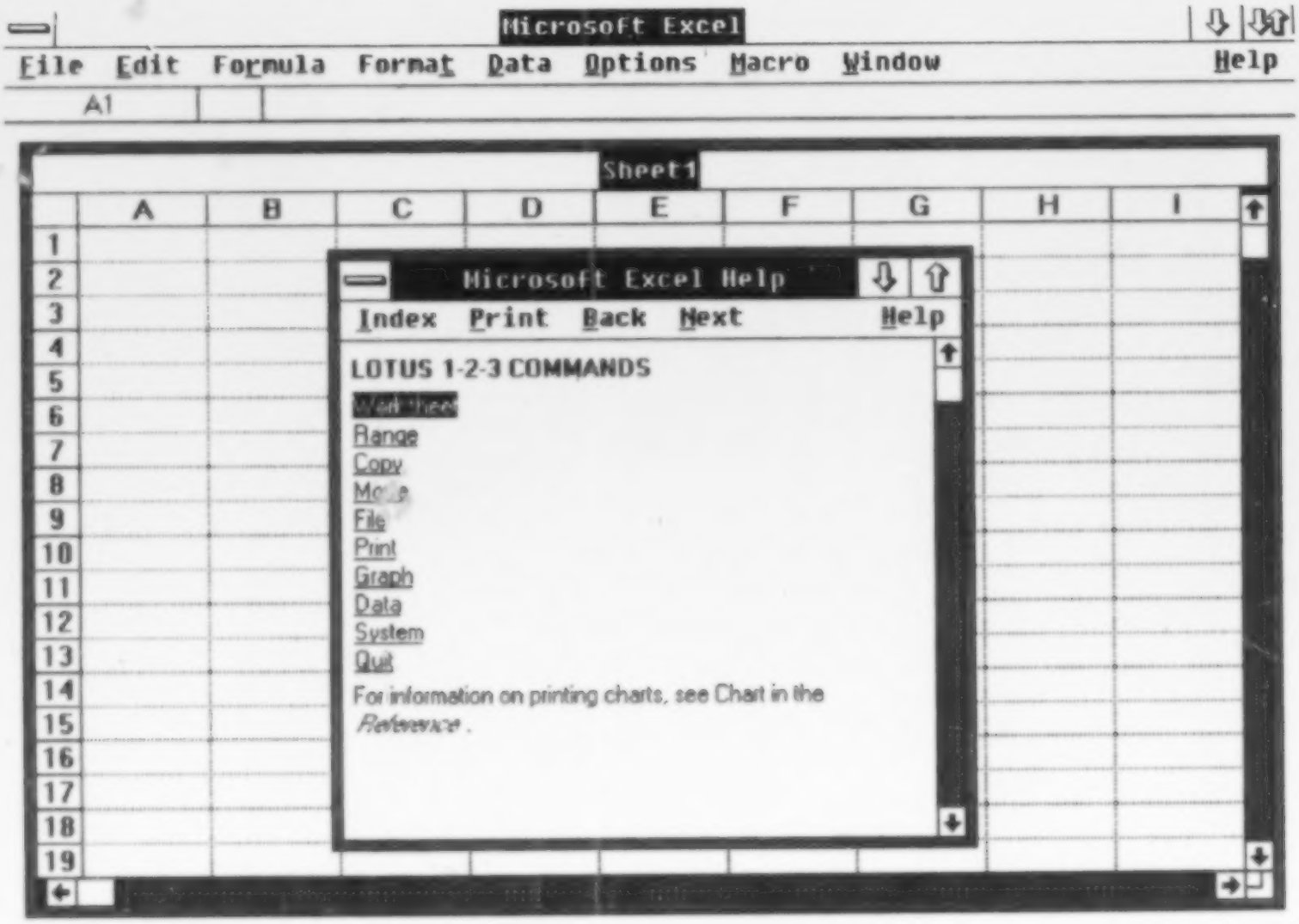
Borland's Quattro Pro Screen Display in "123-Compatible Mode" When Setting the Left Margin for Printing



Lotus 1-2-3 Screen Display, Showing Available Command Choices and Long Prompt Message

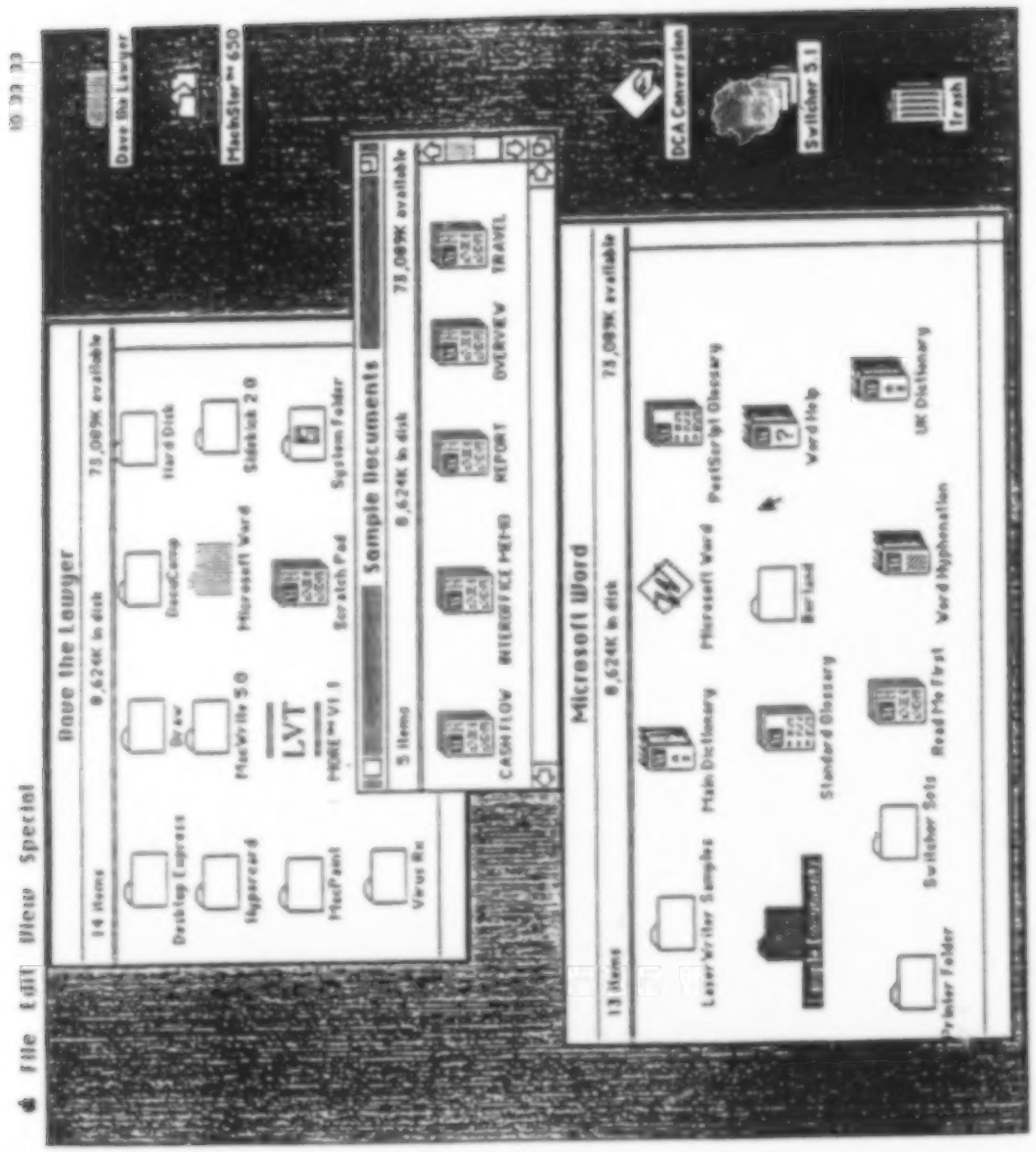


Borland Key Reader, with Blank Box



Microsoft's Excel Screen Display, with Lotus Menus

Macintosh Screen Display



Copyright Act of 1976 (17 U.S.C.)

§ 107. Limitations on exclusive rights: Fair use

Notwithstanding the provisions of sections 106 and 106A, the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—

(1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;

(2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

No. 94-2003

Supreme Court, U.S.
FILED
NOV 9 1995
CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

BRIEF FOR THE PETITIONER

NEAL D. GOLDMAN
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-1127

DONALD J. ROSENBERG
INTERNATIONAL BUSINESS MACHINES CORPORATION
44 South Broadway
White Plains, New York 10601
(914) 288-4010

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

BEST AVAILABLE COPY

62 pp

QUESTION PRESENTED

Whether a computer program's particular menu command hierarchy, which the district court found to contain expression separable from its underlying idea and the functionality it describes, may be protected by copyright in light of the explicit Congressional extension of copyright to computer programs under the same principles applicable to other literary works; or whether, as the First Circuit held, Section 102(b) of the Copyright Act bars protection for any such menu command hierarchy despite its expressive characteristics, because it assists users in communicating with a computer program in order to perform useful operations.

STATEMENT PURSUANT TO RULE 29.6

See pp. i-iv of Petitioner's Reply In Support of Petition for a Writ of Certiorari.

TABLE OF CONTENTS

PAGE

QUESTION PRESENTED i

STATEMENT PURSUANT TO RULE 29.6..... ii

TABLE OF AUTHORITIES vi

OPINIONS BELOW 1

JURISDICTION..... 1

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED 1

STATEMENT OF THE CASE..... 2

A. Procedural History..... 2

B. Factual Background..... 5

1. The nature and purpose of the Lotus 1-2-3 menu command hierarchy 5

2. The creation and value of the Lotus 1-2-3 menu command hierarchy 9

3. The nature and purpose of Borland's copying 11

4. The competitive effect of Borland's copying 15

INTRODUCTION AND SUMMARY OF ARGUMENT..... 16

PAGE

ARGUMENT 19

I. THE COPYRIGHT ACT OF 1976 PROTECTS EXPRESSION IN COMPUTER PROGRAMS UNDER THE TRADITIONAL PRINCIPLES GOVERNING PROTECTION FOR ALL FORMS OF EXPRESSION 19

A. In Section 102(a) of the 1976 Act, Congress Protected the Expression in Computer Programs under Copyright 20

B. In Section 102(b) of the 1976 Act, Congress Applied to Expression in Computer Programs the Same Copyright Principles That it Applied to Other Forms of Expression 26

II. THE EXPRESSION IN THE LOTUS 1-2-3 MENU COMMAND HIERARCHY IS SEPARABLE FROM ITS IDEAS AND FUNCTIONALITY, AND IS THEREFORE ENTITLED TO COPYRIGHT PROTECTION 32

A. The Judicially-Developed Idea/Expression Dichotomy Distinguishes Between the Use of the Ideas Embodied in a Work and Their Particular Description or Explanation 33

B. Under the Idea/Expression Dichotomy, the 1-2-3 Menu Command Hierarchy Contains Protected Expression 39

III. THE FIRST CIRCUIT'S DECISION EFFECTIVELY NULLIFIES THE COPYRIGHT PROTECTION CONGRESS ENACTED FOR THE EXPRESSION IN COMPUTER PROGRAMS 42

PAGE

A. The Decision Below Rests Upon an Interpretation of Section 102(b) Contrary to Congress' Intent 43

B. The Policy Arguments Raised in the Court of Appeals Do Not Justify Depriving Computer Programs of the Statutory Protection to Which They Are Entitled 45

CONCLUSION 50

TABLE OF AUTHORITIES

Cases — PAGE

American Code Co. v. Bensinger, 282 F. 829 (2d Cir. 1922) 25n

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995) 33n, 37n

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) 39n

Atari Games Corp. v. Oman, 888 F.2d 878 (D.C. Cir. 1989) 38

Atari Games Corp. v. Oman, 979 F.2d 242 (D.C. Cir. 1992) 39n

Autoskill Inc. v. Nat'l Educational Support Sys., Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 S. Ct. 307 (1993) 39n

BFP v. Resolution Trust Corp., 114 S. Ct. 1757 (1994) 31

Babbitt v. Sweet Home Chapter of Communities for a Great Oregon, 115 S. Ct. 2407 (1995) 29

Baker v. Selden, 101 U.S. 99 (1879) 33, 34n, 36, 36n, 39

Blunt v. Patten, 3 F. Cas. 763 (C.C.S.D.N.Y. 1828) ... 24n

Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164 (1994) 32n, 48, 49n

College Entrance Book Co. v. Amsco Book Co., 119 F.2d 874 (2d Cir. 1941) 24n

PAGE

Community for Creative Non-Violence v. Reid, 490 U.S. 730 (1989) 29n, 32n, 45-46

Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 36, 36n, 37n

Davis v. Michigan Dep't of Treasury, 489 U.S. 803 (1989) 32

Eastman Kodak Co. v. Image Technical Services, Inc., 504 U.S. 451 (1992) 47n

Emerson v. Davies, 8 F. Cas. 615 (C.C.D. Mass. 1845) 33n

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), *opinion supplemented on denial of reh'g en banc*, 46 F.3d 408 (5th Cir. 1995) 37n

Feist Publications, Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991) 10, 17, 21, 27, 27n, 28, 32, 32n, 41

Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) 33n, 37n

Goodyear Atomic Corp. v. Miller, 486 U.S. 174 (1988) 26n

Gottschalk v. Benson, 409 U.S. 63 (1972) 26

Gustafson v. Alloyd Co., 115 S. Ct. 1061 (1995) 29

Guthrie v. Curlett, 36 F.2d 694 (2d Cir. 1929) 24n

Harper & Row, Publishers, Inc. v. Nation Enter., 471 U.S. 539 (1985) 20n, 47

Hartfield v. Peterson, 91 F.2d 998 (2d Cir. 1937) 25n

PAGE

Hartfield v. Herzfeld, 60 F.2d 599 (S.D.N.Y. 1932) ... 25n

Holmes v. Hurst, 174 U.S. 82 (1899) 29n

Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527 (5th Cir.), *cert. denied*, 115 S. Ct. 82 (1994) 33n

Lasercomb America, Inc. v. Reynolds, 911 F.2d 970 (4th Cir. 1990) 48n

Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 964 F.2d 965 (9th Cir. 1992), *cert. denied*, 113 S. Ct. 1582 (1993) 48n

Mazer v. Stein, 347 U.S. 201 (1954) 20-21, 21, 25, 25n, 46n

Miles v. Apex Marine Corp, 498 U.S. 19 (1990)..... 26n

Mills Music, Inc. v. Snyder, 469 U.S. 153 (1985) 46

Morrissey v. Procter & Gamble Co., 379 F.2d 675 (1st Cir. 1967)..... 33

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931) 37-38, 38n, 40

Parker v. Flook, 437 U.S. 584 (1978) 26

Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487 (2d Cir. 1960) 37

Reiss v. National Quotation Bureau, Inc., 276 F. 717 (S.D.N.Y. 1921)..... 23n

Russello v. United States, 464 U.S. 16 (1983)..... 30n

Ryan v. Royal Ins. Co. of America, 916 F.2d 731 (1st Cir. 1990) 4n

PAGE

Sears, Roebuck & Co. v. Stiffel Co., 376 U.S. 225 (1964) 25

Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) 48n

Stewart v. Abend, 495 U.S. 207 (1990) 20n, 45, 46

Taylor v. United States, 495 U.S. 575 (1990)..... 43n

Two Pesos, Inc. v. Taco Cabana, Inc., 112 S. Ct. 2753 (1992) 48n

United Dictionary Co. v. G. & C. Merriam Co., 208 U.S. 260 (1908) 24n

United States v. Ron Pair Enter., Inc., 489 U.S. 235 (1989) 43n

West Virginia Univ. Hosp., Inc. v. Casey, 499 U.S. 83 (1991) 43

Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 39n

Constitutional Provision

United States Constitution, art. I, § 8, cl. 8.....1, 19, 24

Statutes

15 U.S.C. § 1125(a) 48n

17 U.S.C. § 101 .....2, 22, 27n, 30, 31, 41, 43
§ 102.....*passim*
§ 103 .....2, 27n
§ 106 ..... 20n
§ 107 .....2, 4n, 32n
§ 117 2

PAGE

28 U.S.C. § 1254 1
§ 1292(a)(1) 4

Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124 (repealed 1831) 24n

Copyright Act of 1909, 35 Stat. 1076 (repealed 1976) 20, 21, 21n
§ 3 27
§ 4 21
§ 5 21

Legislative Materials

Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1873 (1974) 23n

Pub. L. No. 96-517, § 12, 94 Stat. 3015 (1980) 23n, 25n

H.R. Rep. No. 1476, 94th Cong., 2d Sess., *reprinted in* 1976 U.S.C.C.A.N. 5659.....21n, 22, 23, 25n, 28

H.R. Rep. No. 1307, 96th Cong., 2d Sess., *reprinted in* 1980 U.S.C.C.A.N. 6460..... 23n

S. Rep. No. 6187, 59th Cong., 2d Sess..... 21n

Rules and Regulations

Fed. R. Civ. P. 52(c)..... 3n

Other

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978).....23-24, 26n, 29, 30, 30n, 31

PAGE

Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559 (1994) 36n, 44n

Peter Huber, "Madonna Ain't Software," *Forbes*, Sept. 3, 1990 .....10n, 47n

David Kahn, *The Codebreakers: The Story of Secret Writing* (1967) 25n

Benjamin Kaplan, *An Unhurried View of Copyright* (1967) 21, 45n

Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977 (1993) 37n, 46n

Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* (1995), § 2.03[D] 26n, 29n
§ 13.03[F][1] 37n, 38n

William F. Patry, *Copyright Law and Practice* (1994) 38n

Charles Selden, *Selden's Condensed Ledger, and Condensed Memorandum Book, and Forms of Record, Condensed Ledger, Reports, and Condensed Memorandum Book* (5th ed. 1861) ... 34n

Lotus Development Corporation ("Lotus") respectfully seeks reversal of the judgment of the United States Court of Appeals for the First Circuit in this case.

OPINIONS BELOW

The opinion of the court of appeals (Pet. App. 1a-28a[1]) is reported at 49 F.3d 807. The opinions of the District Court for the District of Massachusetts (Robert E. Keeton, J.) are reported at 788 F. Supp. 78 (Pet. App. 145a-82a); 799 F. Supp. 203 (Pet. App. 106a-44a); 831 F. Supp. 202 (Pet. App. 71a-105a); and 831 F. Supp. 223 (Pet. App. 29a-70a). The opinion of the district court in the related case of *Lotus Development Corp. v. Paperback Software International* (hereinafter "*Paperback*") is reported at 740 F. Supp. 37 (Pet. App. 183a-269a).

JURISDICTION

The court of appeals entered judgment on March 9, 1995. Pet. App. 1a. This Court granted a petition for certiorari on September 27, 1995. The jurisdiction of this Court is invoked under 28 U.S.C. § 1254.

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED

United States Constitution, art. I, § 8, cl. 8

> The Congress shall have Power . . . [t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors . . . the exclusive Right to their . . . Writings. . . .

---

[1] "Pet. App." refers to the Appendix to Petition for a Writ of Certiorari filed on June 7, 1995, as distinguished from the "JA" or Joint Appendix filed herewith. Items appearing in braces, *e.g.*, "{PX:x;y}," refer to the location in the record where an exhibit, *e.g.*, "PX" for "Plaintiff's Exhibit," was offered ("x") and ruled on ("y") by the district court.

17 U.S.C. § 101, 17 U.S.C. § 102, 17 U.S.C. § 103, 17 U.S.C. § 107, 17 U.S.C. § 117. (The full text of the statutory provisions involved is set forth at Pet. App. 270a-80a and JA 985.)

STATEMENT OF THE CASE

Petitioner Lotus, in the early 1980's, created and began selling the "Lotus 1-2-3" computer program, providing an original expression of a menu command hierarchy for an electronic spreadsheet program to users. Lotus registered copyrights in each version of Lotus 1-2-3 that it published. JA 801-08 {PX 9-12: Dkt. No. 332, 2-52; *id.*}. Respondent Borland International, Inc. ("Borland") later wrote and marketed competing spreadsheet programs called Quattro and Quattro Pro. Borland's programs included what Borland itself called "1-2-3 emulation" menus, which Borland concedes were copied from Lotus 1-2-3.

A. Procedural History

On July 2, 1990, Lotus sued Borland, alleging that Borland infringed Lotus' copyrights by copying protected expression from the Lotus 1-2-3 menu command hierarchy.[2] Upon cross-motions for summary judgment, in July 1992 the district court granted Lotus' motion in part. The court held that the 1-2-3 menu command hierarchy contained expressive elements protected by Lotus' copyrights and that Borland had copied that expression, but reserved certain issues for trial concerning the precise extent of the 1-2-3 menu command hierarchy's copyrightability.[3]

[2] The suit was commenced four days after Lotus received a favorable ruling on a similar copyright claim in *Paperback*. The day after the *Paperback* decision, Borland filed a declaratory judgment action seeking a declaration of non-infringement in the Northern District of California, No. C-90-20386 VRW. JA 1. The California district court dismissed that action in favor of this one. *Id.* at 2.

[3] Pet. App. 116a. The district court also granted Lotus' motion for summary judgment dismissing Borland's affirmative defense of waiver,

Following the summary judgment decision, Borland removed the visible "1-2-3 emulation" menus from its products and began to publicize the existence of a feature called the "Key Reader," which was derived from those menus. *Id.* at 54a-56a; JA 785, 919 {PX 43: Dkt. No. 404, 5-67; *id.*}. Lotus was granted leave to file a supplemental complaint alleging that the Key Reader also infringed its 1-2-3 copyrights. JA 6 (Dkt. No. 305).

The district court held two bench trials after Borland withdrew its jury demand. Pet. App. 75a-77a. The scope of the Phase I trial was defined by stipulation as "all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto," excluding Key Reader issues. JA 40-45. The Phase II trial addressed all issues relating to the Key Reader. *Id.* at 48-50. The district court issued separate opinions concerning the two phases of trial.

In the Phase I opinion, the court held that the "1-2-3 emulation" menus copied a qualitatively substantial amount of protected expression from the 1-2-3 menu command hierarchy (which it called the "menu structure"). Pet. App. 83a-94a. The district court also dismissed Borland's estoppel and laches defenses to the original complaint, based upon extensive and detailed findings of fact.[4] The Phase II opinion held that the

but held that trial was necessary concerning its defenses of estoppel and laches. *Id.* at 141a (waiver), 141a-42a (laches) & 142a-43a (estoppel).

[4] *Id.* at 95a-105a. Having failed to plead the affirmative defense of fair use in its answer to the original complaint, Borland first asserted the defense in closing argument at the Phase I trial over Lotus' objection. JA 107-17. The district court granted Borland leave to amend its answer to the original complaint *after* trial. *Id.* at 56-63 (motion); Dkt. No. 353, dated March 30, 1993 (granting motion). The district court then granted judgment in Lotus' favor pursuant to Fed. R. Civ. P. 52(c), rejecting Borland's argument that its infringement had no actual or potential negative effect on the value of Lotus' copyrights. JA 187-92; Pet. App. 76a-77a. Borland's contentions were based entirely upon three Lotus statements indicating that, at various times, Lotus had held the same share of the DOS spreadsheet market. JA 183.

Key Reader also contained a "virtually identical" copy of "details of expression of the Lotus 1-2-3 program's menu structure." Pet. App. 46a-47a. The district court further rejected all of Borland's affirmative defenses to the supplemental complaint.[5] Concluding that Borland had infringed Lotus' copyrights, the district court entered a permanent injunction (at Borland's request) on August 19, 1993, prohibiting further sales of Borland's products in a form that contained the infringing features. *Id.* at 69a-70a. Borland appealed to the First Circuit as of right pursuant to 28 U.S.C. § 1292(a)(1). JA 9 (Dkt. No. 408).

Oral argument was initially heard on April 6, 1994, and the case was reargued on October 6, 1994, in the wake of then Chief Judge (now Justice) Breyer's recusal. JA 11. The court of appeals issued its opinion reversing the judgment of the district court on March 9, 1995 (*id.* at 12), four days before a damages trial was scheduled to commence. *See* Dkt. No. 668. The First Circuit held that the 1-2-3 menu command hierarchy was not entitled to copyright protection.[6] Lotus filed a petition for a writ of certiorari on June 7, 1995, which was granted on September 27, 1995.

---

[5] *Id.* at 48a-69a. Regarding fair use, the district court carefully considered the factors specified in 17 U.S.C. § 107, finding that each weighed in Lotus' favor. *Id.* at 60a-68a. In particular, the district court found that "Borland's limited evidence concerning market share is entirely insufficient to demonstrate that Lotus has not been harmed" by Borland's use. *Id.* at 67a.

[6] Pet. App. 22a. The First Circuit therefore did not address any of the affirmative defenses Borland raised on appeal. In addition to fair use, Borland argued in the First Circuit only the defenses of laches to the original complaint and waiver to the supplemental complaint. Brief of Defendant/Appellant Borland International, Inc., at 58-63 (1st Cir.) (No. 93-2214). Accordingly, its other affirmative defenses were abandoned. *Ryan v. Royal Ins. Co. of America*, 916 F.2d 731, 734 (1st Cir. 1990).

B. Factual Background

In the court of appeals, Borland conceded—and the court accepted—that Borland copied the Lotus 1-2-3 menu structure or menu command hierarchy, in its entirety, in the Borland Quattro and Quattro Pro programs.[7] We focus, accordingly, on facts pertinent to determining the copyrightability of the 1-2-3 menu command hierarchy as a whole, in particular its nature, purpose, and original creation. We then turn to the facts surrounding Borland's incorporation of the 1-2-3 menu command hierarchy in its products, and demonstrate that the factual premises upon which the First Circuit relied in reaching its decision are either contrary to, or unsupported in, the record below.

1. The nature and purpose of the Lotus 1-2-3 menu command hierarchy.

A proper understanding of the nature and purpose of the 1-2-3 menu command hierarchy is critical to an appropriate resolution of this case. As used by the First Circuit, the term refers to 469 words or "menu commands" arranged in more than 50 different "menus" and sequential sub-menus, which are displayed to the user on the computer monitor screen during use of the Lotus 1-2-3 computer program. Pet. App. 4a. The menus appear in variable series designed by the authors of 1-2-3 to respond to users' choices from previous menus, branching out from the first or "main" menu in what often is

---

[7] Pet. App. 10a. Lotus proffered direct proof of Borland's copying in the district court (JA 549-55, 574-75, 707-08, 763-66, 778 & 794-96), and an examination of the products themselves reveals that no other conclusion is possible. *See id.* at 64-69 (Dkt. Nos. 359, 360) (the computer programs at issue, installed on a computer, are part of the record herein, together with the products' user's manuals and packaging) (PX 1-8, 20-27, 35, 50, DX 516, 518-521: Dkt. No. 332, 2-[illegible] 2-16; Dkt. No. 333, 3-26:4-11); *id.* at 317-19 (Dkt. No. 107, Ex. A) (books containing photographs of the menus at issue) (Dkt. No. 332, 2-35, 2-36; Dkt. No. 333, 3-26:4-11); Dkt. No. 118, Ex. thereto (videotape demonstrating the programs in operation) (Dkt. No. 332, 2-35, 2-36; Dkt. No. 333, 3-26:4-11).

called a "menu tree." *Id.* at 78a-84a; JA 326, 370 & 375. It is the copyrightability of the overall combination of words and menus in the 1-2-3 menu command hierarchy, viewed as a whole, and *not* any individual menu command such as "COPY" or "PRINT," viewed in isolation, that is at issue in this case.[8]

Despite the First Circuit's comparison of the menu commands to the buttons on a videocassette recorder ("VCR") machine (Pet. App. 18a), the words do not, in themselves, perform any mechanical function. They are simply words of text. These words describe—when read in sequence—an array of operations available to the user and, like the pages of an instruction manual or reference guide, provide information to assist users in selecting the appropriate instructions to cause the program to perform certain tasks. Pet. App. 78a-81a; JA 333-36, 377. They are, and were intended to be, a form of structured dialogue between 1-2-3's authors and users, in which the capabilities of the program are explained in plain English and users, in turn, can learn which instructions will cause the program to perform particular functions by making choices from successive menus.

Lotus 1-2-3 is an example of a type of computer program known as a "spreadsheet" application program.[9] Typically, spreadsheet programs provide users with the ability to perform arithmetical and mathematical operations on numerical data

[8] Similarly, Lotus' claim is based upon the content that Borland copied from 1-2-3's screens, including the words, their order, and overall arrangement, rather than the color, style, layout, or format in which the words or menus are displayed on the screens.

[9] Pet. App. 186a; JA 375. Personal computer programs generally are divided into two categories: operating systems and application programs. Pet. App. 186a. The former control the basic functions of the computer, such as the internal allocation of computer memory. *Id.* The latter permit a user to perform a set of related operations directed towards particular tasks, such as word processing or database management. *Id.* Typically, application programs are designed to work with a particular operating system, for example, DOS or UNIX, and will not work with a computer directly.

entered by the user in an electronic representation of a spreadsheet grid, arranged in columns and rows; to reconfigure the style or layout of the spreadsheet to suit the user's particular needs; and to manipulate and present the data in the spreadsheet in a variety of different tables or graphs of the user's choice. Pet. App. 226a-27a.

The Lotus 1-2-3 menu command hierarchy is part of what is typically called the "user interface" of the program, because it is the program component with which the user directly communicates. *Id.* at 226a-29a; JA 324, 333, 360 & 375. The user interface is not free-standing, however. It is part of the program, and is generated by the statements contained in the program's "source code."[10] As the district court explained in *Paperback*, computer programs typically are written in source code, using a form of programming language, such as BASIC or FORTRAN, that employs symbolic names and rules of syntax to express instructions for the computer to execute.[11] The user usually never sees or even is aware of the program code during operation of the program. However, the actual contents of the user interface—including the words appearing in the 1-2-3 (or Borland) menus—is spelled out, in text, in the program code. JA 64 {PX 13: Dkt. No. 332, 2-52, 2-53; Dkt. No. 333, 3-26-4:11}, 360, 548 & 756-58.

In order to learn or recall the appropriate instructions to give the program to perform functions that correspond to the

[10] JA 330, 450. In the case law, user interfaces often have been referred to as "non-literal elements" of computer programs. Pet. App. 258a-60a.

[11] Pet. App. 188a-89a. The source code must be converted or "compiled," typically by a distinct "compiler" program, into a binary form that the digital computer can recognize, called a machine language, before the computer can execute the program. Pet. App. 189a. Compiled source code is called "object code." *Id.* Most commercial software is distributed only in the form of object code.

menu commands,[12] the user presses the "*/*" (slash) key on the computer keyboard. Pet. App. 123a, 227a. The program, in response, displays the main menu of ten words, beginning with "WORKSHEET," "RANGE," "COPY," and "MOVE."[13] Most of these do not relate to a specific operation that the program will perform. Rather, they describe categories of commands, and serve to guide the user to another menu, or sub-menu, of increasingly specific command choices, and so on until all the instructions necessary to cause the program to perform a particular task have been described. *Id.* at 78a-84a; JA 334-35, 384-85 & 735.

At each menu level, the user indicates a selection from the menu by striking one of the cursor (arrow) keys to highlight the particular menu command and then striking the "Enter" key, or by striking a letter key corresponding to the first letter of the word, *e.g.*, "W" for "WORKSHEET."[14] Only after an entire sequence of keystrokes is completed (and, usually, after some data is entered, such as an identification of the portion of the spreadsheet to be affected), does the program

---

[12] The menu commands are not the only commands available to users when working with 1-2-3, nor do they cause the program to perform arithmetical or mathematical operations. Such operations are invoked by typing "formulas" within the individual "cells" in the spreadsheet grid, using familiar arithmetical notation (*e.g.*, "+" or "-") or special commands called "@ functions" (because they are preceded by the "@" symbol) for more complex mathematical expressions. Pet. App. 232a. Only the menu commands are at issue in this case.

[13] *Id.* at 227a-28a. The description in text is of the versions of Lotus 1-2-3 that Borland copied, all of which were published prior to November 1986. *Id.* at 81a, 96a & 113a.

[14] Pet. App. 227a-28a; JA 376. This particular style of menu, called the "moving cursor menu," was conceived by Lotus' founder, Mitchell Kapor, and employed in two earlier programs he had written. Pet. App. 286a. Lotus has never contended that the idea of a moving cursor menu, in contrast to the words in the menus that give them meaning and content, is protected by Lotus' copyrights. The district court so held in *Paperback* (*id.* at 230a, 254a), and it is not at issue here.

actually perform an operation involving the spreadsheet.[15] The words appearing in the menus on the screen—what the user perceives—therefore can be distinguished from the physical tasks—the keystrokes—the user must undertake to send an instruction to the program, as well as from the operation that the program ultimately performs after the instruction is sent.

2. The creation and value of the Lotus 1-2-3 menu command hierarchy.

The 1-2-3 menu command hierarchy was created in the latter half of 1982 by Mr. Kapor and a team working under his direction. Pet. App. 231a. Kapor and his team previously had specified the selection of functional capabilities that the program would provide (such as allowing the user to replicate data or formulas entered in one portion of a spreadsheet in another), and had implemented those capabilities in the program's code. *Id.* at 288a. The menu commands reflected Kapor's subsequent attempt to express those capabilities to users, in full words "that would intelligently convey to the user the purpose of each command." *Id.* at 291a. The arrangement of the menus in a hierarchical structure was intended to communicate "the product's underlying functionality" to users in a clearly organized presentation. *Id.* Kapor and his team spent hundreds of hours over a period of many months considering (and discarding) dozens of organizations of the menu tree, and refining the choice of each word in the menus and the order of those words within each menu.[16] The menu hierarchy Kapor ultimately selected "was based largely on my intuition and subjective judgment . . . trying as best I could to imagine myself in the

---

[15] JA 334-35, 384-85. Thus, it is not true that the user can issue an instruction described by a single menu command, such as striking the key "P" for "PRINT," and cause the program to perform a printing operation, as the First Circuit appeared to believe. Pet. App. 15a.

[16] *Id.* at 290a-96a; JA 538-39. Kapor and his team were not only free to choose any existing word, but also could invent new words, as they did in at least one instance (*i.e.*, "Xtract"). Pet. App. 295a.

role of a typical user." *Id.* at 292a. He had no rule book, model, or data to guide him.[17]

It is undisputed that the 1-2-3 menu command hierarchy is original for copyright purposes, in that Lotus created it independently, without copying or derivation from prior works. *See Feist Publications, Inc. v. Rural Telephone Serv. Co.*, 499 U.S. 340, 345 (1991) ("Original, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity."). The district court in *Paperback* held that the 1-2-3 menu command hierarchy was "very different" from previous products. Pet. App. 234a. Borland never asked the district court to rule differently in this case.[18] Even aside from originality in the narrow copyright sense, the creativity expressed in the 1-2-3 menu command hierarchy, by making the program's capabilities clear and comprehensible to users, formed a major part of the program's value.[19]

---

17 *Id.* 1-2-3 also was designed to allow the user to create "macros," or sequences of frequently-used keystrokes that can be executed automatically with a single keystroke. *Id.* at 290a. Macros may include commands other than menu commands, and keystrokes having nothing to do with commands at all. *Id.* at 228a-29a.

18 Although Borland did suggest that the 1-2-3 menus were not novel because some of their words had been used previously in other programs or in programming languages, that is not—and never has been—relevant under copyright law. Borland never contended, and could not prove, that the words used in Lotus 1-2-3's menu command hierarchy had appeared previously in a *single* program, arranged in the same hierarchical organization, to describe the same set of functional operations. JA 314-15.

19 *See* Peter Huber, "Madonna Ain't Software," *Forbes*, Sept. 3, 1990, at 104 ("unlike the alphabet, command structures are not a universal and ancient heritage of the spreadsheet business; they are what distinguish the good, the bad and the ugly . . . . Some, like the Lotus spreadsheet, are wonderfully transparent and intuitive.").

In a world in which graphical user interfaces for personal computer software have become commonplace, it may be difficult to appreciate the extent to which the Lotus 1-2-3 menu command hierarchy represented a major advance in user interface design when the product was introduced in January 1983. At the time, the user interfaces of popular programs were relatively crude. For example, VisiCalc—the leading spreadsheet program prior to 1-2-3—displayed a list of letters arranged in alphabetical order to represent its array of available commands. Pet. App. 233a, 267a. This cryptic notation forced users to memorize the commands, or to consult the product's manuals or other written documentation, in order to decipher the letters and take advantage of the program's capabilities. Despite significant established competition, Lotus 1-2-3 soon supplanted VisiCalc as the most popular DOS spreadsheet program.[20] Its user interface—including its menus—played "a substantial role" in that success, as the district court found. Pet. App. 135a; JA 474. Borland's own software developers admitted that menus convey information to users and assist them in learning the program. JA 563-64, 568-70, 580-81, 589-90, 767-68, 769 & 782-83. Both sides' experts further agreed that the 1-2-3 menus possessed "great commercial significance."[21]

3. The nature and purpose of Borland's copying.

Borland did not need to copy the Lotus 1-2-3 menu command hierarchy to create a spreadsheet product that would provide the same functional capabilities as 1-2-3. As the dis-

---

20 Competitive products available in 1983 included VisiCalc as well as Microsoft MultiPlan, SuperCalc, Context MBA, and numerous others. Exhibit A to the Declaration of Jim P. Manzi, dated November 1, 1991 (Dkt. No. 111).

21 *Id.* at 333, 343-44, 454, 507 & 697-98. Although Borland contended that 1-2-3 would have succeeded just as well if it had used other, equally good, words or arrangements, the district court was not required to resolve that issue at any stage of the proceedings below, and it did not purport to do so.

trict court found (and the First Circuit accepted), the 1-2-3 menu command hierarchy is only one of a very large number of possible ways to express a set of instructions a user can give to a spreadsheet program. Pet. App. 86a-88a, 131a-33a. The record before the district court contained numerous examples of contemporaneous programs that performed the same functions, but employed different menu hierarchies and menu commands. *Id.* at 88a; JA 64-70, 339. Indeed, the district court found that, even if a software developer voluntarily restricted itself to designing a program that provided *exactly* the same set of functional capabilities that 1-2-3 provided, using the same type of hierarchical menu structure, there still would be millions of possible permutations for an acceptable spreadsheet menu command hierarchy. Pet. App. 131a-32a; JA 338. Other words could be chosen to represent each menu command; the order of the words within each menu could be altered; and the placement of commands in particular menus—*i.e.*, the decisions as to which commands should depend upon other commands, at what depth in the hierarchical arrangement—could be varied. Pet. App. 86a, 131a-32a; JA 337-38, 378-79, 506, 699-702 & 719-22.

Borland copied the 1-2-3 menu command hierarchy, therefore, and included it in its products for commercial reasons, not as a technical necessity. Simply put, it sought to minimize its risk in entering the spreadsheet market by exploiting 1-2-3's prior success. Borland was not the first Lotus competitor to pursue this tack. Lotus' early success was followed rapidly by imitators such as Paperback Software, who sought to enter the spreadsheet market with self-described 1-2-3 "clones," or programs displaying menu command hierarchies deliberately copied from Lotus 1-2-3, which sold at substantially lower prices. Pet. App. 147a-48a, 236a-38a. Lotus commenced the *Paperback* case in January 1987. *Id.* at 96a. With full awareness of that pending case, Borland released the first version of its spreadsheet products, called "Quattro," in November 1987, and a second product, called "Quattro Pro,"

in November 1989. *Id.* at 96a, 100a. These products also were offered to 1-2-3 users at less than half the retail price of 1-2-3. JA 907-14 {PX 15, 18-19: Dkt. No. 332, 2-57, 2-58; Dkt. No. 333, 3-26:4-11}.

Quattro and Quattro Pro differed from earlier "clones" chiefly by providing alternative menu command hierarchies that the user could select: a so-called "native" menu tree, which was developed independently by Borland and differed materially both in menu content and in hierarchical arrangement from 1-2-3 (JA 559-60, 770-72 & 779); and the "1-2-3 emulation" menu tree, which contained a virtually identical copy of the Lotus 1-2-3 menu command hierarchy. Pet. App. 82a. Both types of menu trees allowed users to invoke the same set of functional capabilities. *Id.* at 131a; JA 774-75. Contrary to the lower court's impression that it "takes some effort" to "access[ ] the Lotus menu in the Borland" products (Pet. App. 25a), Borland made it easy for users to find and select the "alternative" 1-2-3 menus.[22] Once those menus were selected, they would appear automatically each subsequent time the program was run.[23]

Although Borland's designers naturally believed their own menus were superior to 1-2-3's (JA 558-60, 579, 770-71 & 779), Borland nevertheless decided to include the 1-2-3 menu command hierarchy in its products to enhance its marketing

---

[22] The user's manuals Borland sold with its products prominently described the specific commands required to select a menu tree. *See, e.g.*, JA 65-66 (*Getting Started with Quattro* at 4, ch. 4 at 23 ("Quattro for 1-2-3 Users"); *Getting Started with Quattro Pro* at 2, ch. 4 at 27 ("Quattro Pro for 1-2-3 Users")). The user needed merely to choose the so-called "123.RSC" or "123.MU" options from the Quattro and Quattro Pro menus, respectively, to cause the program to display the 1-2-3 menus whenever the "slash" key was struck. *Getting Started with Quattro* at 26; *Getting Started with Quattro Pro* at 38-39. *See also* JA 705 (Borland expert opining on the ease of switching menu trees).

[23] JA 65-66 (*Quattro User's Guide* at 152-53; *Quattro Pro User's Guide* at 196-97), 705.

efforts. Borland assumed that existing 1-2-3 users would be disinclined to "learn a new menu tree" and therefore might be reluctant to purchase its products. *Id.* at 739-40. The "1-2-3 emulation" menus were seen as a "migration tool," which would allow Borland to overcome this resistance by promoting the presence of the "familiar and comfortable" 1-2-3 menu tree as an alternative to Borland's own. *Id.* at 558, 568-70, 595, 652-54, 739 & 741-42. The experts on both sides agreed that this provided Borland with a significant marketing advantage (*id.* at 330-33, 354, 704-05 & 731-32), which Borland tried to exploit at every opportunity. Its promotional material stressed: "If you know how to use 1-2-3, you know how to use Quattro[.] You don't have to learn a whole new program."[24]

Borland also promoted its products to 1-2-3 users by stressing that they could preserve and reuse the macros they had written for use with 1-2-3. Borland described this as "macro compatibility," which it marketed along with "menu compatibility" as among its chief competitive advantages. JA 595-96 ("we promoted everything that we could think of that would make Quattro easier for Lotus users to adopt"). *See also id.* at 66 (product packaging for first version of Quattro Pro), 737-38 (promotional activities). Originally, Borland users could run 1-2-3 macros only by employing the "1-2-3 emulation" menus in Borland's products. *Id.* at 568-71, 764-66 & 778. After this lawsuit commenced, Borland devised a new feature in order to continue to provide "macro compatibility" without the visible display of the 1-2-3 menus. *Id.* at 784-86. Called the Key Reader publicly—but described as the "phantom 1-2-3 menus" inside Borland—this consisted of a stripped-down version of the same program file Borland had used to display the "1-2-3 emulation" menus, which was now

[24] *Id.* at 908. *See also id.* at 909 ("*Exceptional compatibility with Lotus® 1-2-3®*. Use the same commands and macros as you would in 1-2-3.") (emphasis in original).

hidden inside the program. Pet. App. 30a-33a; JA 787-94, 915 {PX 32: Dkt. No. 404, 5-67; *id.*}, 916 {PX 33: Dkt. No. 404, 5-67; *id.*} & 931-67 {DX 520 [sic]: Dkt. No. 403, 4-164; *id.* at 4-200}.

4. The competitive effect of Borland's copying.

The court of appeals rested its decision on a number of factual premises and assumptions that were not tried or proven in the district court concerning competition in the spreadsheet market. For example, both the majority and concurring opinions appear to assume that, if Borland had not copied the 1-2-3 menu command hierarchy, spreadsheet users would have been "locked in" to Lotus because of their unwillingness to learn or invest in a new program. Pet. App. 20a, 26a. Yet, while Borland's own executives uniformly testified at deposition that inclusion of the 1-2-3 menus was an "important" factor in such commercial success as Borland did achieve, they could not swear that it was *necessary* to allow Borland to offer a commercially viable product. JA 544-46, 590, 594, 653, 657-58 & 740-41. The effect of Borland's copying on either its or Lotus' sales was never litigated or adjudicated in the district court. Nor was it proven in the district court that Borland's "success" was "due primarily to other features," or to inherent product superiority, as both the majority and concurring opinions below assume.[25] What caused consumers to purchase Borland's products, or even who those consumers were, was not before the court of appeals and could not properly form a basis for its decision.

[25] Pet. App. 4a, 26a-27a. The First Circuit's views might be explained by the fact that the district court, in its infringement analysis, necessarily compared the versions of 1-2-3 that Borland copied—which were written in 1986 or before—to Borland's newer 1987 and 1989 products, which contained the copied menus. The court of appeals apparently did not realize that there were newer versions of Lotus' products as well, and that it was not able to compare Borland's products to the contemporaneous versions of 1-2-3 with which they actually competed.

The First Circuit also appeared to assume that market conditions had not changed from 1987, when Borland first decided to copy the Lotus 1-2-3 menus. By 1993, however, both Lotus and Borland had fallen far behind another competitor, Microsoft (and its product Excel), as the personal computer software market shifted away from DOS products to the Microsoft Windows operating environment. Indeed, when Borland removed the "1-2-3 emulation" menus from its products following the district court's summary judgment decision in July 1992, it described the 1-2-3 menu command hierarchy as an "outdated user interface." JA 919. The court below simply ignored the dynamic nature of the market, assuming that Lotus in 1995 continued to hold "such sway" that it "represented the *de facto* standard for electronic spreadsheet commands." Pet. App. 26a. In the software industry, however, "standards" may change, as the histories of both VisiCalc and Lotus 1-2-3 have shown. The First Circuit's views concerning the market effects of its decision thus rested on an incomplete and deeply flawed understanding of the competitive terrain, derived from a nonexistent record.

INTRODUCTION AND SUMMARY OF ARGUMENT

The issue in this case is whether the copyright protection Congress granted computer programs protects the original, creative expression contained in their user interfaces against wholesale appropriation by those who determine to achieve commercial success through imitation rather than innovation. As we show below, Congress provided in the Copyright Act of 1976, and reaffirmed in the 1980 Software Amendments, that computer programs are to be protected under copyright as literary works, subject to the same principles applicable to other, more familiar works of authorship. Applying the traditional tools of copyright analysis to new forms of expression has not always been a simple or straightforward task, but the courts have successfully faced such challenges before.

Indeed, until the First Circuit ruled in this case, the various courts of appeals were well on their way toward reaching consensus concerning the new issues presented by copyright in computer programs. That the application of old doctrines to new technologies may not always be easy, however, is no excuse for courts to disregard a clear Congressional mandate.

In this case, the First Circuit denied all copyright protection to a new form of expression—the Lotus 1-2-3 menu command hierarchy—because it did not appreciate (or understand) the work and ultimately did not believe that it should be protected. This case is not about VCR buttons, automobile gearshifts, or any other type of machine part.[26] It is about a collection of hundreds of words that the authors of 1-2-3 carefully arranged in a complex structure to express to computer users an original way to communicate with a particular kind of computer program. Were these words fixed upon a series of printed pages, instead of appearing on a computer screen, there could hardly be a question that they would be protected by copyright. Even a shorter and simpler work could contain the minimal creative spark that copyright requires. *Feist*, 499 U.S. at 358-59.

Why, then, did the First Circuit reach a different outcome in this case? Its decision provides no sound answer. Beyond a facile and flawed analogy to VCR buttons,[27] the sum and

---

[26] As Judge Keeton wrote in *Paperback*, while "similes, metaphors, and other forms of allusion are appropriately a part of our efforts to communicate ideas," they are no substitute for "logically compelled inferences from authoritative declarations." Pet. App. 242a.

[27] The problem with the VCR button analogy is *not* simply the fact that a computer program is a copyrightable work, while a VCR is not, thus rendering a claim for copyright protection of the "user interface" of the latter inherently suspect. Although valid, this is not the most important distinction. Rather, on a more fundamental level, a purported "work of authorship" consisting of six buttons labeled "Stop," "Pause," "Play," "Fast Forward," "Rewind," and "Eject," would surely fail to qualify as copyrightable under one or more of the traditional copyright tests such

substance of the First Circuit's analysis is its determination to treat the words "method of operation" in Section 102(b) of the 1976 Copyright Act as disqualifying, *per se*, all elements of a work that can be defined as *part* of a "method of operation," including expressive elements. By means of this definitional short-cut, the First Circuit could dispense with the abstract and sometimes difficult job of separating the "idea" (or "method" or "process") in a work from any protected expression it may contain. The district court, in contrast, undertook this task and, after extensive fact-finding and analysis, concluded that the 1-2-3 menu command hierarchy did contain separable expression worthy of protection. Having chosen a more simplistic rule of decision, the First Circuit dismissed the district court's efforts as irrelevant.

Whether this approach was driven by a misguided search for a bright line test in an area of law that has never yielded any, or instead by the policy considerations articulated in both the majority and the concurring opinions below, it is clear that the First Circuit's analysis cannot be reconciled with Congress' manifest intention to grant copyright in computer programs. Congress stated that computer programs should receive protection as literary works, notwithstanding their utilitarian and functional nature, and despite its understanding that they are used to implement a process or method. If, as the First Circuit held, anything that can be defined as part of a method of operation is ineligible for copyright protection, then by the same logic virtually everything in a computer program—source code no less than textual menus on the screen—is unprotected. This result effectively negates Congress' intent. Moreover, Congress explicitly stated, and this Court has held, that Section 102(b) is a codification of

---

as merger, *scenes a faire* or originality. Lotus 1-2-3's menu command hierarchy, on the other hand, passed each of these tests—based upon a full trial record—as would a table of contents, or an abbreviated user's manual, containing 469 textual entries arranged in more than 50 hierarchically-linked pages.

the "idea/expression dichotomy." This compels exactly the kind of fact-based, line-drawing exercise that Judge Keeton faithfully undertook, but the First Circuit never attempted.

Ultimately, the First Circuit's decision rests not upon the statute, legislative history, or precedent—which it disregarded without more than a passing reference or attempt at distinction—but upon a cluster of subjective policy notions and a mistaken belief that it was free to decide what the law should be. Even if the record below provided an accurate and informed basis for making such policy determinations—and it does not—the First Circuit simply failed to heed this Court's repeated admonitions that, under the Constitution, such judgments are for Congress to make, not the courts. In this case, Congress did speak; the court below failed to listen.

ARGUMENT

I

THE COPYRIGHT ACT OF 1976 PROTECTS EXPRESSION IN COMPUTER PROGRAMS UNDER THE TRADITIONAL PRINCIPLES GOVERNING PROTECTION FOR ALL FORMS OF EXPRESSION

The outcome of this case depends upon the proper interpretation of Section 102 of the Copyright Act of 1976 (the "1976 Act"), 17 U.S.C. § 102, in light of Congress' decision to recognize meaningful copyright protection for original works of authorship expressed in computer programs. This case does *not* rest solely on an interpretation of Section 102(b), as the First Circuit appeared to believe. Pet. App. 17a. That subsection can be understood only in the context of the entirety of Section 102 and the remainder of the statute.

Congress' authority to confer copyright protection is found in the United States Constitution, art. I, § 8, cl. 8:

> The Congress shall have Power . . . [t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors . . . the exclusive Right to their . . . Writings.

Congress exercised this power when it enacted Section 102 of the 1976 Act, which grants copyright protection in an extremely broad range of subject matter:

> Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced or otherwise communicated, either directly or with the aid of a machine or device. (17 U.S.C. § 102(a).)

Copyright protection is a form of property awarded by Congress to achieve the objectives of the constitutional clause. The statute vests a bundle of distinct property rights in the author from the time of the work's creation[28]—most notably in this case, the right to prevent others from copying or reproducing the work without permission.

A. In Section 102(a) of the 1976 Act, Congress Protected the Expression in Computer Programs under Copyright.

Congress deliberately employed expansive language in Section 102(a) of the 1976 Act in an attempt to avoid the misunderstanding that arose concerning the scope of copyrightable subject matter under the Copyright Act of 1909, 35 Stat. 1076 (the "1909 Act"), and to allow for the foreseeable emergence of new media of expression. Before 1909, Congress attempted to accommodate new forms of expression by specifically enumerating the categories of works entitled to copyright protection, gradually expanding the scope of coverage. *Mazer v.*

---

[28] *Stewart v. Abend*, 495 U.S. 207, 220 & n.3 (1990); *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 539, 546-47 (1985). *See* 17 U.S.C. § 106 (listing various exclusive rights).

Stein, 347 U.S. 201, 208-10 (1954).[29] Congress rejected this approach in the 1909 Act.[30] Thus, Congress defined the "works for which copyright may be secured" in Section 4 of the 1909 Act as "all the writings of an author," with the intention of including "all forms of record in which the thought of an author may be recorded and from which it may be read or reproduced." *Id.* (internal quotation omitted). Despite this definition, however, the list of registration categories set forth in the succeeding Section 5 was, "as a practical matter . . . read to limit the general statement." Benjamin Kaplan, *An Unhurried View of Copyright* 39 (1967). *See Feist*, 499 U.S. at 352 (noting that some courts misread Section 5 of the 1909 Act). Confusion arose because the legislative history of the 1909 Act also suggested that Section 4 was "declaratory of existing law" only. *Mazer*, 347 U.S. at 210.

Acting in a context of increasingly rapid technological change, Congress enacted Section 102(a) of the 1976 Act to replace Section 4 of the 1909 Act. *Feist*, 499 U.S. at 355. Congress' embracive intent is clear in its language: *any* "original work[ ] of authorship" that is "fixed in *any* tangible medium of expression," which can be perceived or communicated directly *or* "with the aid of a machine or device," is copyrightable. By enacting protection for expression in all media "now known or later developed," Congress demonstrated its intention not to limit the works eligible for copyright to the forms then in existence or in popular use, but to protect new forms of expression that would become possible only through future technological advances.[31]

---

[29] By 1904, Congress had enacted "some twenty-five laws dealing with copyrights." *Id.* at 209 n.12.

[30] The legislative history of the 1909 Act reflects Congress' belief that the "existing statutes attempt specifications which are unfortunate because necessarily imperfect and requiring frequent additions to cover new forms or new processes." S. Rep. No. 6187, 59th Cong., 2d Sess., at 4.

[31] *See* H.R. Rep. No. 1476, 94th Cong., 2d Sess., at 51 (the "House Report"), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5664 ("The history of

Section 102(a) goes on to provide a list of eight categories of protected "works of authorship," which are intended to be "illustrative and not limitative." House Report at 53, *reprinted in* 1976 U.S.C.C.A.N. at 5666. The legislative history demonstrates that Congress did not intend these categories to

> exhaust the scope of "original works of authorship" that the bill is intended to protect. Rather, the list sets out the general area of copyrightable subject matter, but with sufficient flexibility to free the courts from rigid or outmoded concepts of the scope of particular categories. *Id.*

One such protected category is "literary works," defined in Section 101 of the 1976 Act as works, "other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects . . . in which they are embodied." 17 U.S.C. § 101. Both a computer program's code and its textual menus are expressed in such "words, numbers, or other verbal or numerical symbols or indicia," fitting the literal terms of the definition. Both are fixed in a tangible medium of expression and can be "perceived, reproduced, or otherwise communicated," either directly or with the aid of a computer or other "machine or device." Both therefore comprise copyrightable subject matter under Section 102(a). Indeed, the legislative history of the 1976 Act confirms that Congress specifically intended that computer programs, as well as textual works (such as data bases) communicated through the medium of a computer, should be protected as "literary works":

---

copyright law has been one of gradual expansion in the types of works accorded protection . . . . Authors are continually finding new ways of expressing themselves, but it is impossible to foresee the forms that these new expressive methods will take. The bill does not intend either to freeze the scope of copyrightable subject matter at the present stage of communications technology or to allow unlimited expansion into areas completely outside the present congressional intent.").

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. *It also includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.* (House Report at 54, *reprinted in* 1976 U.S.C.C.A.N. at 5667 (emphasis supplied).)

Congress' enactment was well within its Constitutional power.[32]

In 1980, Congress reaffirmed this intention when it adopted, without material modification, the recommendations of the National Commission on New Technological Uses of Copyrighted Works ("CONTU").[33] Congress had established CONTU in 1974 for the purpose, *inter alia*, of studying the relationship between computers and copyrights, and making recommendations concerning what, if any, changes in the law were appropriate to accommodate computer programs.[34] Following extensive study, public hearings, and debate, CONTU recommended that computer programs continue to receive protection under the 1976 Act as "literary works." *See* Final Report of the National Commission on New Technological

---

[32] As Learned Hand wrote in *Reiss v. National Quotation Bureau, Inc.*, 276 F. 717, 719 (S.D.N.Y. 1921) (holding a cable and telegraph code book copyrightable), the Constitution "is not a strait-jacket, but a charter for a living people," and grants Congress the power to protect not only forms of expression known in 1789 but also "what the ingenuity of men should devise thereafter."

[33] Pub. L. No. 96-517, § 12, 94 Stat. 3015, 3028 (1980) (codified at 17 U.S.C. §§ 101, 117) (the "1980 Software Amendments"). *See* H.R. Rep. No. 1307, 96th Cong., 2d Sess. at 23, *reprinted in* 1980 U.S.C.C.A.N. 6460, 6482 (the pertinent section of the bill "embodies the recommendations of [CONTU] with respect to clarifying the law of copyright of computer software").

[34] Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1873-74 (1974).

Uses of Copyrighted Works (1978), at 1-2 & 37-46 (hereinafter "CONTU Report").

In recognizing copyright protection for computer programs, Congress did not depart from traditional principles, as the court of appeals appeared to believe. Pet. App. 24a-25a. Congress was well aware that computer programs serve a useful—as well as an expressive—purpose. But that has never been a bar to copyrightability. Indeed, the Constitution itself declares that the goal of copyright is to "promote the Progress of Science and *useful* Arts." United States Constitution, art. I, § 8, cl. 8 (emphasis supplied). And the First Congress extended copyright protection to maps and charts, which may possess aesthetically pleasing attributes when considered as antiques or decorations, but are created for a utilitarian purpose.[35]

Copyright traditionally has protected such useful types of "writing" as catalogs, dictionaries, and various forms of instructional or reference materials.[36] Moreover, long before 1976 it was well established that commercial cable and telegraph codes—consisting of arbitrary symbols and words intended only to serve the useful purpose of enabling economical and confidential business communications—were copyrightable, when embodied in the tangible medium of code books necessary to decipher their meaning.[37] As this Court stated in *Mazer v. Stein*, 347 U.S. at 218, with regard to the copyrightability of statuettes used as lamp bases: "We find nothing in the copyright statute [of 1909] to support the argument that the intended use or use in industry of an article eligible for copyright bars or invalidates its registration."

Nor can it be presumed—as the concurring opinion in the court below apparently does—that Congress failed to consider legislating under patent law, rather than copyright law, in exercising its power to protect the utilitarian expression in computer programs. Pet. App. 23a-25a. As the Court held in *Mazer*, the two bodies of law are *not* mutually exclusive.[38] Both are based upon the same Constitutional authority. *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225, 228-29 (1964). However, when Congress passed the 1980 Software Amendments,[39] this Court had twice held that computer programs,

---

35 Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124 (repealed 1831). *See Blunt v. Patten*, 3 F. Cas. 763, 764 (C.C.S.D.N.Y. 1828) (accepting copyrightability of plaintiff's navigational charts under Act of 1790, but finding no infringement when defendant's charts were created from own surveys and public sources and were not copied from plaintiff's).

36 *See, e.g., United Dictionary Co. v. G. & C. Merriam Co.*, 208 U.S. 260 (1908) (noting copyrightability of Webster's High School Dictionary); *College Entrance Book Co. v. Amsco Book Co.*, 119 F.2d 874 (2d Cir. 1941) (reversing dismissal of case for infringement of book containing lists of French words to be studied for the New York Board of Regents examination, based upon the "originality" in plaintiff's choice of words, articles, and translations); *Guthrie v. Curlett*, 36 F.2d 694, 696 (2d. Cir. 1929) (holding copyrights valid in consolidated freight tariff index because author "expressed" idea of consolidating relevant information "in the form and arrangement he chose to use" and "[t]he thought, arrangement and style was original," but finding no infringement when defendant had not copied the means of expression used by the author).

37 *See Hartfield v. Peterson*, 91 F.2d 998, 999, 1000 (2d Cir. 1937) (A. Hand, J.) (finding infringement when "phrases" in defendant's code book were copied from plaintiff's copyrighted code book); *American Code Co. v. Bensinger*, 282 F. 829, 832 (2d Cir. 1922) (sustaining copyright in a book of code words and symbols used "to enable telegraphic messages to be transmitted with simplicity, economy, and secrecy"); *Hartfield v. Herzfeld*, 60 F.2d 599, 599-600 (S.D.N.Y. 1932) (restraining infringement of artificial code words published in a book with parallel "Wall Street" expressions). *See also* David Kahn, *The Codebreakers: The Story of Secret Writing* 802-853 (1967) (describing historical development and usage of commercial cable and telegraph codes).

38 347 U.S. at 217. Congress was not only aware of *Mazer* when it passed the 1976 Act, it also acted affirmatively to incorporate its holding in the statute. House Report at 55, *reprinted in* 1976 U.S.C.C.A.N. at 5667-68.

39 The 1980 Software Amendments were included as Section 10 of a bill entitled "To amend the patent and trademark laws," which also revised, in preceding sections, various provisions of patent law. *See* Pub. L. No. 96-517, 94 Stat. 3015 (1980).

standing alone, did not comprise patentable subject matter. *Gottschalk v. Benson*, 409 U.S. 63 (1972); *Parker v. Flook*, 437 U.S. 584 (1978). Congress chose not to disturb these rulings, opting instead to provide incentives to stimulate progress in this relatively new field of creative endeavor through the vehicle of copyright.[40] There is simply no basis in the statute or its legislative history to support the notion that Congress failed to understand the decision it made, or to make that decision after considering the alternatives.

B. In Section 102(b) of the 1976 Act, Congress Applied to Expression in Computer Programs the Same Copyright Principles That it Applied to Other Forms of Expression.

Contrary to the First Circuit's view that Section 102(b) enumerates categories of works "foreclosed" in their entirety from copyright protection (Pet. App. 21a), that provision merely serves to limit the extent to which copyright will protect a work that is copyrightable under Section 102(a) from unauthorized copying.[41] Section 102(b) provides:

> In no case does copyright protection for an original work of authorship *extend* to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work. (17 U.S.C. § 102(b) (emphasis supplied).)

[40] *See* CONTU Report at 16-18 (comparing copyright and patent protection; noting uncertain state of law concerning patentability of computer programs and likely difficulties faced by applicants for program patents). *See also Miles v. Apex Marine Corp*, 498 U.S. 19, 32 (1990) ("We assume that Congress is aware of existing law when it passes legislation."); *Goodyear Atomic Corp. v. Miller*, 486 U.S. 174, 184-85 (1988) (same).

[41] *See* 1 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* (1995), § 2.03[D] at 2-34 (hereinafter *Nimmer*) (the distinction "constitutes not so much a limitation on the copyrightability of works, as it is a measure of the degree of similarity which must exist between a copyrightable work and an unauthorized copy, in order to constitute the latter an infringement").

The phrasing of Section 102(b) presupposes the existence of an "original work of authorship" entitled to "copyright protection," and simply makes it clear that the scope of that protection does not "extend" to any "idea, procedure, process, system, method of operation, concept, principle, or discovery" that the work reveals, regardless of how it is embodied in the work. Section 102(b) therefore explicitly distinguishes between an "idea" and the description or explanation of that idea. The "copyright protection" that "subsists in" a work by reason of Section 102(a) includes the particular expression of an "idea" in the work, but under Section 102(b) does not extend to the "idea" itself.[42]

The legislative history confirms that this is what Congress intended. Section 102(b) replaced Section 3 of the 1909 Act, which stated that copyright in a work protected only "the copyrightable components of a work." *Feist*, 499 U.S. at 355-56. This provision was thought to be ambiguous. *Id.* at 351. Section 102(b) thus "identifies specifically those elements of a work for which copyright is not available," but does so in a way that was not intended to alter the distinction that courts previously had drawn between an unprotected "idea" and the copyrightable "expression" of that idea (*id.* at 356):

> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal sys-

[42] This interpretation of Section 102 comports with this Court's interpretation of Section 103 of the 1976 Act. Section 103—concerning "compilations," or selections and arrangements of preexisting facts or data (*see* 17 U.S.C. § 101 (definition of "compilation"))—contains a structure similar to Section 102. As this Court has explained, Section 103(a) affirmatively states that "the subject matter of copyright . . . includes compilations," while Section 103(b) makes it clear that copyright in such a work "protects only the author's original contributions—not the facts or information conveyed." *Feist*, 499 U.S. at 359.

> tem of copyright, that the basic dichotomy between expression and idea remains unchanged. (House Report at 57, *reprinted in* 1976 U.S.C.C.A.N. at 5670.)

In *Feist*, this Court observed that the "idea/expression" or "fact/expression" distinction in Section 102(b) "applies to all works of authorship" (499 U.S. at 350)—a reading grounded in the words of the provision itself. 17 U.S.C. § 102(b) ("*In no case* does copyright protection for an original work of authorship extend . . . .") (emphasis supplied). And the legislative history confirms that Congress intended that Section 102(b) be interpreted so that the scope of copyright protection for computer programs is determined under the same principles that apply to other types of copyrightable works:

> Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law. (House Report at 57, *reprinted in* 1976 U.S.C.C.A.N. at 5670.)

Congress plainly understood that computer programs had a functional aspect, and were used to accomplish a "process" or "method of operation." Yet, like a compilation that can both "embody" the idea of a collection of facts and express that idea in an original selection and arrangement of those facts, Congress recognized that a computer program can both "embody" a process or method *and* contain a "copyrightable element" in the particular "expression adopted by the programmer" to implement that same process or method.[43]

---

[43] The Court, as well, long has understood that the same text may both reveal an "idea" and contain copyrightable expression in the author's original arrangement of words selected to describe that idea. *Holmes v. Hurst*, 174 U.S. 82, 86 (1899).

Congress' intention that courts should interpret the terms "process" and "method of operation" in Section 102(b) in a manner consistent with traditional principles, and not in some manner unique to computer programs or other utilitarian works, is shown by their inclusion in a list of undifferentiated terms following the word "idea"—a term with a settled meaning in the copyright field.[44] Under the doctrine of *noscitur a sociis*, a word in a statute "gathers meaning from the words around it." *Babbitt v. Sweet Home Chapter of Communities for a Great Oregon*, 115 S. Ct. 2407, 2415 (1995), *quoting from Jarecki v. G.D. Searle & Co.*, 367 U.S. 303, 307 (1961). *See Gustafson v. Alloyd Co.*, 115 S. Ct. 1061, 1069 (1995) (single word in a statute should not be given "a meaning so broad that it is inconsistent with its accompanying words"). Thus, applying the term "method of operation" in Section 102(b) to computer programs, copyright protects the particular words or symbols selected by the programmer to express a method (like the words chosen to express an idea), but not the "method of operation" (or "idea") itself.[45]

Again, Congress confirmed this interpretation of Section 102(b) by adopting CONTU's recommendations in the 1980 Software Amendments. CONTU realized that computer programs could be characterized as comprising a "process" or "method of operation." CONTU Report at 18-20. Neverthe-

---

[44] *See Community for Creative Non-Violence v. Reid*, 490 U.S. 730, 739 (1989) (applying "well established" principle that when Congress uses terms that have achieved settled meaning under the common law, a court must infer that Congress intends to incorporate the established meaning of those terms, unless the statute otherwise dictates).

[45] *See* 1 *Nimmer*, § 2.03[D] at 2-35 ("It would, then, be a misreading of Section 102(b) to interpret it to deny copyright protection to 'the expression' of a work, even if that work happens to consist of an 'idea, procedure, process, etc.' Thus, if a given 'procedure' is reduced to written form, this will constitute a protectible work of authorship so as to preclude the unlicensed copying of 'the expression' of the procedure, even if the procedure per se constitutes an unprotectible 'idea.' ").

less, the Commission agreed with Congress' earlier judgment that the "idea/expression" distinction was sufficient to guide the courts to draw an appropriate line for copyright protection of computer programs. It declared: "That the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability."[46]

CONTU also proposed the inclusion of the following definition of a "computer program":

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result. (*Id.* at 12.)

By incorporating this definition in Section 101 of the 1976 Act, Congress left no doubt of its intention to accord copyright protection to the particular "set of statements or instructions" expressed in a computer program, despite the fact that they would be "used" in implementing "method[s] of operation" to accomplish functional "result[s]." Moreover, because both a program's code and its menu command hierarchy express a "set of statements or instructions" that can be used to accomplish "a certain result" in a computer, they both fit the definition of a "computer program" that Congress adopted.

Had Congress intended to limit the copyright protection computer programs were to receive under Section 102, it knew how to do so. Congress confronted a somewhat analogous issue in its treatment of "useful articles," which are

---

[46] *Id.* at 21. CONTU did recommend one unique restriction on copyright protection for computer programs not material to the issues before the Court, proposing that owners of a legitimate copy of a program receive a limited right to make another copy or adaptation if necessary. CONTU Report at 12. That Congress enacted CONTU's one proposed limitation, but no other, demonstrates that Congress did not intend to treat programs differently from other forms of "literary works" in any other respect. *See, e.g., Russello v. United States*, 464 U.S. 16, 22 (1983) (noting that if Congress had intended to restrict the scope of a subsection of the RICO statute, "it presumably would have done so expressly" as it did elsewhere in the statute).

defined in Section 101 as articles "having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." 17 U.S.C. § 101. "Useful articles" are copyrightable as "pictorial, graphic, and sculptural works"—one of the illustrative categories of works set forth in Section 102(a)—"only if, and only to the extent that, [their] design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101. Congress could have defined "computer programs" in a similarly qualified fashion. It did not, instead defining them as "literary works"—creating an inference that no such limitation was intended. *See BFP v. Resolution Trust Corp.*, 114 S. Ct. 1757, 1761 (1994) (when Congress includes language in one section of a statute but omits it in another, courts presume that Congress acted purposely in enacting the disparate provisions).[47]

Thus, it is clear that Congress, in enacting both the 1976 Act and the 1980 Software Amendments, did *not* intend that the scope of protection for expression in computer programs should be governed by any special rules or limitations, despite its awareness that programs are utilitarian works that may be used for essentially functional purposes. Rather, Congress specifically intended the expression in computer programs to enjoy copyright protection under the same principles governing other forms of original expression in literary works. Moreover, neither the statute nor the legislative history suggests that Congress intended to treat differently those words or symbols the program displays to users in a menu command

---

[47] Although Congress chose *not* to classify computer programs as "useful articles," and although menus are not "useful articles" because their purpose is merely "to convey information," Judge Keeton did draw such an analogy (Pet. App. 117a), and at the trial in this case he applied the standard applicable to "useful articles" to test the copyrightability of the 1-2-3 menu command hierarchy. He found that it was, in fact, conceptually separable from, and was not "functionally dictated" by, the operations of the program that it described. Pet. App. 92a-94a.

hierarchy from those within its source or object code. The definitions Congress adopted for "literary works" and "computer programs" apply to each equally well. Congress' decision was within its Constitutional authority and fully consistent with the long-established tradition of recognizing copyright protection in the expressive elements of works that may also serve a useful purpose.

II

THE EXPRESSION IN THE LOTUS 1-2-3 MENU COMMAND HIERARCHY IS SEPARABLE FROM ITS IDEAS AND FUNCTIONALITY, AND IS THEREFORE ENTITLED TO COPYRIGHT PROTECTION

As this Court held in *Feist*, 499 U.S. at 350, Congress intended in Section 102(b) to codify the judicially-developed "idea/expression" dichotomy, which courts traditionally have applied to determine the scope of protection to be accorded to copyrighted works. "When Congress codifies a judicially defined concept, it is presumed, absent an express statement to the contrary, that Congress intended to adopt the interpretation placed on that concept by the courts." *Davis v. Michigan Dep't of Treasury*, 489 U.S. 803, 813 (1989).[48]

[48] Section 102(b) is only one of several provisions in which Congress codified a judicially defined concept in the 1976 Act. *Reid*, 490 U.S. at 749 n.15. *See Feist*, 499 U.S. at 355 (stating that Congress did not intend in Section 102(a) to contract the bounds of copyrightable subject matter); *Campbell v. Acuff-Rose Music, Inc.*, 114 S. Ct. 1164, 1170 (1994) (holding that Congress intended the "fair use" provision of Section 107 of the 1976 Act "to restate the present judicial doctrine of fair use, not to change, narrow, or enlarge it in any way") (quotation omitted).

A. The Judicially-Developed Idea/Expression Dichotomy Distinguishes Between the Use of the Ideas Embodied in a Work and Their Particular Description or Explanation

The idea/expression dichotomy obliges courts to determine whether the particular element of a copyrighted work that an accused infringer has copied is merely an "idea" embodied in the original work, or is a particular way to express that idea chosen by the author from a universe of many possibilities. When the element that was copied is one of only a few conceivable ways to express the idea, or is at a level of abstraction so general that to deem it an infringement effectively precludes all others from publishing any other expression of that idea, the idea and expression are said to "merge" and copyright protection is foreclosed. *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967).[49]

The origin of the "idea/expression" dichotomy and the related merger doctrine generally is traced to this Court's decision in *Baker v. Selden*, 101 U.S. 99 (1879), although—like "fair use"—its roots run deep in the common law of copyright.[50] Selden published a series of pamphlets intro-

[49] *See also Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir. 1994) (noting "well-recognized" precept that when an idea and its expression are indistinguishable, or "merged," the expression will be protected only against nearly identical copying), *cert. denied*, 115 S. Ct. 1176 (1995); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 533 (5th Cir.) (noting that, when idea can be expressed in very few ways, copyright law does not protect that expression; "In such cases idea and expression are said to be merged."), *cert. denied*, 115 S. Ct. 82 (1994); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 838 (10th Cir. 1993) ("Under the merger doctrine, copyright protection is denied to expression that is inseparable from or merged with the ideas, processes, or discoveries underlying the expression.").

[50] *See, e.g., Emerson v. Davies*, 8 F. Cas. 615, 618-19 & 625 (C.C.D. Mass. 1845) (Story, J.) (holding that plaintiff's original "plan, arrangement and combination of materials" for a series of arithmetic lessons was copyrightable, even if the materials so arranged did not originate with the author, and finding infringement when defendant copied

ducing a self-described "new system of Book-keeping" intended primarily for use by municipal treasurers, which contained a preface page, a one-page introductory essay, a collection of forms illustrating the practical use of the "system," and another collection of blank forms.[51]

Baker devised a competing "system" of ledger forms that—as his counsel argued in this Court—used a different "arrangement" of columns and differed markedly from Selden's in other respects:

> The entries in Baker's are upon the principle known as single entry; the entries in Selden's are known as double entry. The situation of columns is reversed and other differences shown by the testimony cited above are so radical as to entirely relieve the Baker system from the charge of infringements. (Argument for Appellant at 16 & 18, *Baker v. Selden* (No. 95, October Term, 1879).)

The only commonality lay in the use of certain column headings required of county treasurers by Ohio statute (*id.* at 10-13), together with the use of ruled columns in a system based upon the "same generic principles," which would "produce the same result" of "keep[ing] the different accounts accurately."[52]

Observing that the "evidence of the complainant is principally directed to the object of showing that Baker uses the

that plan and arrangement instead of composing a new arrangement from the same "common materials and common sources of knowledge, open to all men").

51 Charles Selden, *Selden's Condensed Ledger, and Condensed Memorandum Book, and Forms of Record, Condensed Ledger, Reports, and Condensed Memorandum Book* (5th ed. 1861) (copies of which are lodged with the Court). The pamphlets are described as "books" in the Court's opinion because, at the time they were registered, that was the statutory category of copyrightable subject matter to which such works belonged. *Baker*, 101 U.S. at 101.

52 *Id.* at 16 & 18. Comparison of Baker's ledger forms (Transcript of Record at 119-143, *Baker v. Selden*), with Selden's reveals few, if any, literal similarities.

same system,"[53] the Court concluded that Baker's system "uses a similar plan so far as results are concerned; but makes a different arrangement of the columns, and uses different headings." 101 U.S. at 100. Thus, the Selden work was not infringed "in any way, unless the latter became entitled to the exclusive right in the system." *Id.* at 101. The Court then rejected the contention that copyright gave Selden the "exclusive right to the use of the system or method of book-keeping" as illustrated in his forms, or in forms "made and arranged on substantially the same system." *Id.* As the Court declared:

> [W]here the art [a book] teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public; not given for the purpose of publication in other works explanatory of the art, but for the purpose of practical application. (*Id.* at 103.)

Conversely, the Court stated that copyright *does* protect the author's particular *statement* of the "rules and methods of useful art" as "embodied and taught" in a copyrighted work:

> The use by another of the same methods of statement, whether in words or illustrations, in a book published for teaching the art, would, undoubtedly, be an infringement of the copyright . . . . The use of the art is a totally different thing from a publication of the book explaining it. (*Id.* at 104.)

53 Selden's estatrix argued that Selden's copyrights "extend to and embrace a *system of bookkeeping*—a combination of lines so arranged as to suggest an improved method of classifying and condensing mercantile accounts. The lines are the symbols to convey the idea of the method or plan." Argument for Appellee at 6-7, *Baker v. Selden* (emphasis in original).

Thus, the Court concluded that the copyright in Selden's "book" of forms "did not confer upon him the exclusive right to make and use account-books ruled and arranged as designated by him and described and illustrated in said book." *Id.* at 107.

Baker v. Selden thus establishes the basic idea/expression dichotomy and the related merger doctrine. Copyright will not prevent the practical use by the public of the "art" a work teaches, or of illustrations and forms that are "necessary incidents to the art"—*i.e.*, that reflect a "merger" of idea and expression. But this limitation does not permit the unlicensed use of the author's description, explanation, or illustration of the art in another work "published for teaching the art." An element of a work, therefore, does not lose all copyright protection merely because it describes a system or method that its author intended for "people" to learn and use, as the First Circuit erroneously concluded from its reading of *Baker*. Pet. App. 18a. People may be free to use the method, but no one can publish another work describing or explaining that method using the author's particular expression of it.

To articulate the holding of *Baker v. Selden* is easier than to apply its principle. As the Second Circuit has written, the decision "offers scant guidance on how to separate idea from expression," or on how to "distinguish protectible expression from that expression which 'must necessarily be used as incident to' the work's underlying concept." *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2d Cir. 1992).[54] Applying the idea/expression dichotomy to the facts of a particular case has always presented a challenge, not only in cases involving computer programs.[55] To determine whether

[54] As the district court noted in this case, the Second Circuit's *Altai* decision takes *Baker* as the "starting point" for its analysis, but did not hold that *Baker* controlled the outcome: "The Second Circuit was sensitive not only to its duty of fidelity to precedent, but as well to its duty of fidelity to congressional mandates that came into existence long after *Baker* was decided." Pet. App. 127a.

[55] *See* Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559, 2560 (1994) (disagreement over identification of the expression that may be protected without risk of monopolizing the underlying idea "is hardly peculiar to computer programs"); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 1035 (1993) ("Courts always have had to struggle with the delicate questions of where the commonplace becomes originality and where legitimate imitation shades into infringement, especially in the environment of a new mode of expression.").

particular expression is inseparable from or "merges" with any idea or method it embodies is, in all events, a fact-specific inquiry, which defies any "bright line" rules. After five decades on the bench, no less a jurist than Learned Hand could declare only that decisions as to when an imitator has gone beyond the idea and borrowed the expression of a work "must inevitably be *ad hoc*." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960). The distinctions required are the sort of subtle, yet important and necessary, judgments that courts are routinely called upon to make in copyright cases—as in many others.

Although the methodologies courts have employed vary somewhat in their details, a consensus has emerged around the so-called "abstraction-filtration-comparison" test for applying the idea/expression distinction in the context of computer programs.[56] The test draws upon the "patterns of abstraction" analysis articulated in *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930) (L. Hand, J.), *cert. denied*, 282

[56] *See, e.g., Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1342-43 (5th Cir. 1994) ("Generally, we endorse the abstraction-filtration-comparison method of determining copyright protection for computer programs . . . "), *opinion supplemented on denial of reh'g en banc*, 46 F.3d 408 (5th Cir. 1995); *Apple Computer v. Microsoft*, 35 F.3d at 1445-46 (endorsing "dissection" of nonliteral elements of computer programs to determine infringement and noting that abstraction-filtration-comparison test performs same analysis, although articulated differently); *Gates Rubber*, 9 F.3d at 840-41 (accepting basic three-part analysis of abstraction-filtration-comparison test articulated in *Altai*); *Altai*, 982 F.2d at 706-11. *See also* 3 *Nimmer*, § 13.03[F][1] at p. 13-125-50 (describing "successive filtration" test).

U.S. 902 (1931), to identify an appropriate conception of the work's unprotected ideas.[57] Typically, a "filtration" step is then applied, in which the elements of the copyrighted work are analyzed to screen out those that are unprotected for other reasons.[58] Finally, the court compares the protected elements of the copyrighted work with the accused work to determine whether infringement has occurred.

Whether any particular circuit's version of this test, or even another type of test, is preferable from a doctrinal or practical standpoint is not at issue in this case.[59] Every court of appeals that has addressed this issue (other than the First Circuit) has reached the common understanding that Section 102(b) requires courts to attempt to separate the purely functional attributes of computer programs from the particular expression chosen by the programmer to accomplish or to provide that functionality. *See Atari Games Corp. v. Oman*, 888 F.2d 878, 884-86 (D.C. Cir. 1989) (R.B. Ginsburg, J.) (reversing judgment affirming Copyright Office refusal to register copyright in video game, for failing to explain how the work was "dictated by 'functional requirements' " and lacked expression "separable from the game itself" that would

[57] In *Nichols*, Judge Hand recognized that infringement could occur without literal copying of a work's text, or "else a plagiarist would escape by immaterial variations." 45 F.2d at 121. The "abstraction" analysis seeks to identify the point in the "patterns of increasing generality" that may be discerned in a work where the statement of what has been copied is no more than an unprotected "idea" (*id.*)—*e.g.*, "a play about star-crossed young lovers," "a system for keeping municipal accounts," or "a spreadsheet program with a hierarchical menu structure." This insight continues to furnish a durable construct for idea/expression analysis.

[58] These filters are simply applications in this particular context of the traditional copyright doctrines of merger, originality, and *scenes a faire*. 3 *Nimmer*, § 13.03[F][1] at p. 13-130 & 13-135 n.303.13.

[59] *See* 1 William F. Patry, *Copyright Law and Practice* 225-26 (1994) (suggesting that courts should focus less on devising tests and more on interpreting the statute, including the statutory definitions, to determine the materiality of an alleged infringer's taking).

"qualify as copyrightable subject matter").[60] To employ the language of *Baker v. Selden*, courts must attempt to separate the practical *use* of the system from the author's particular *statement* explaining or describing the system. Prior to the First Circuit's decision in this case, the circuits had reached consensus that, when such a separation is possible, the expression is protected by copyright if it is only one of numerous possibilities for providing the same functionality.[61] The First Circuit, alone, has seen fit to interpret Section 102(b) in a contrary manner.

B. Under the Idea/Expression Dichotomy, the 1-2-3 Menu Command Hierarchy Contains Protected Expression

The outcome in this case does not depend upon which articulation of the idea/expression distinction is employed—as Borland conceded in oral argument in the court of appeals, and as that court apparently accepted. Pet. App. 14a-15a.

[60] *See also Atari Games Corp. v. Oman*, 979 F.2d 242, 247 (D.C. Cir. 1992) (R.B. Ginsburg, J.) (reversing second refusal to register copyright in same video game, when Copyright Office failed to consider scope of expression in work as a whole under the threshold standard of creativity enunciated in *Feist*).

[61] *See, e.g., Autoskill Inc. v. Nat'l Educational Support Sys., Inc.*, 994 F.2d 1476, 1494 (10th Cir.) (upholding lower court's refusal to apply merger doctrine to deny protection to elements of computer program, when that court found no evidence that the program's "idea" could be expressed in only one way), *cert. denied*, 114 S. Ct. 307 (1993); *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 & n.28 (3d Cir. 1986) (finding structure of computer program to be part of its expression, because "idea" of efficiently managing dental laboratory could be accomplished in a number of different ways with a number of different programs; "Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea."), *cert. denied*, 479 U.S. 1031 (1987); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983) ("If other methods of expressing that idea are not foreclosed as a practical matter, then there is no merger."), *cert. dismissed*, 464 U.S. 1033 (1984).

Indeed, the First Circuit dismissed as "immaterial" to its analysis *any* consideration of the extent to which the selection and arrangement of the 469 words comprising the 1-2-3 menu command hierarchy reflected " 'expressive' choices," or was capable of alternative expression. Pet. App. 17a. But these are precisely the questions a court *must* consider in order to determine whether a particular element of a work is protected by copyright. Judge Keeton engaged in detailed fact-finding on these issues in the district court upon an ample trial record, and found that Borland had copied protected expression in the 1-2-3 menu command hierarchy. The First Circuit did not disturb or reject these findings as clearly erroneous; it simply chose to ignore them.

The district court's test for determining whether Borland had copied protected expression in the Lotus 1-2-3 menu command hierarchy was based upon one it devised in the earlier *Paperback* case (*id.* at 216a-17a), refined in light of the issues presented in this case. *Id.* at 163a-65a, 170a-71a. Like the "abstraction-filtration-comparison" test, Judge Keeton's began with a *Nichols* "abstraction" analysis (*id.* at 118a-20a, 163a, 219a-222a), proceeding to a second step in which elements of the work that are "functionally dictated" or are "necessary incidents" to the idea, system, or method, were removed.[62] Applying this test to the 1-2-3 menu command hierarchy, the district court considered five possible conceptions of its underlying idea and found that the following formulation was the proper one:

> [Its] user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the

[62] *Id.* at 119a-20a, 127a, 163a-65a. Judge Keeton's test also concluded with a determination of the qualitative substantiality of the copied expression. *Id.* at 119a, 135a, 163a-65a, 171a.

> specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy. (Pet. App. 73a-74a.)

No lesser level of abstraction is possible without defining the "idea" of the 1-2-3 menu command hierarchy as itself—a tautology the district court found to be unjustified. *Id.* at 130a. This was so because, under the district court's definition, another spreadsheet program could offer *exactly* the same selection of functions as does 1-2-3 without infringing, so long as it used different expression in its menus. *Id.* Thus, the district court concluded that the Lotus 1-2-3 menu command hierarchy contained "expression" distinguishable from its "idea" (or "system") and was entitled to protection.

The district court also found, after trial, that the 1-2-3 menu command hierarchy was not determined by its underlying functionality (*id.* at 84a-90a), and that the "text of the menu commands and the menu structure itself are not dictated by mechanical application of the functional considerations" Borland contended had confined its expression. *Id.* at 92a. Judge Keeton went on to apply the principles used to find separable expression in "useful articles," even though the statutory definition of "useful articles" does not include computer programs. *See* 17 U.S.C. § 101. Judge Keeton found that the functional operations of the program could be conceptually separated from the text and arrangement of the menus explaining them, and that this latter expression "easily satisfies" the originality requirement as articulated by this Court in *Feist*.[63]

[63] Pet. App. 92a. Concerning the Key Reader, the district court was careful to limit its decision *only* to the specific issues raised by Borland's use of "phantom" 1-2-3 menus to run 1-2-3 macros. *Id.* at 39a. In particular, Judge Keeton found that Borland had implemented the Key Reader by stripping all but the first letters from "the same copy of the 1-2-3 menu structure and commands that Borland had used in its emulation interface," in what he called a "stripped menu tree." *Id.* at 35a. Borland nonetheless had copied protected expression because the Key Reader contained a recognizable copy of the 1-2-3 menus. *Id.* at 44a. Like a compilation that expresses, in its arrangement, an original way to

The First Circuit dispensed with the need to review the district court's fact-based conclusion by reading Section 102(b) as a definitional filter, rather than as a codification of the idea/expression dichotomy. That is, once the court of appeals decided to affix a pejorative label to the 1-2-3 menu command hierarchy, no further analysis was necessary. The court may have spared itself the difficult task of finding the line between idea and expression by ruling as a matter of law that *all* menus are *per se* uncopyrightable methods of operation—no matter how much creative, original, and separable expression they contain—but it did so at the expense of fidelity to an uninterrupted line of precedent and despite Congress' clear intent to the contrary.

III

THE FIRST CIRCUIT'S DECISION EFFECTIVELY NULLIFIES THE COPYRIGHT PROTECTION CONGRESS ENACTED FOR THE EXPRESSION IN COMPUTER PROGRAMS

The First Circuit fundamentally misread Section 102(b) of the 1976 Act as automatically barring copyright protection for any aspect of a computer program that can be described as embodying a "method of operation," regardless of the expressive elements it contains.[64] This interpretation has no support in the language of the 1976 Act, the 1980 Software Amendments, or the legislative history. It also runs directly against the entire body of decisional authority before and after the

gain access to uncopyrightable facts, the district court found that the arrangement of the 1-2-3 menu tree expresses an original way to specify the set of executable (and uncopyrightable) operations in a spreadsheet program. *Id.* at 40a-41a.

[64] Pet. App. 17a. The First Circuit essentially rewrote Section 102(b) to state: "In no case does copyright protection extend to any method of operation, *or* to any description, explanation, illustration, or embodiment thereof, in an original work of authorship."

passage of the 1976 Act, which consistently has applied the idea/expression distinction to determine the scope of protection for all copyrighted works. The First Circuit's ruling effectively nullifies Congress' decision to recognize copyright protection for computer programs and fails to accord to owners of copyright in such works the same scope of protection for their property that owners of copyright in other works of authorship enjoy.

A. The Decision Below Rests Upon an Interpretation of Section 102(b) Contrary to Congress' Intent

The First Circuit described the Lotus 1-2-3 menu command hierarchy as a "method of operation" because, in that court's view, it is the "means by which a person operates something"—*i.e.*, a computer—in order to achieve a functional result.[65] But the same can be said of computer programs generally, because all computer programs can be described as "the means by which a person operates" a computer—as the concurring opinion below acknowledged. *Id.* at 24a. Congress understood that computer programs were "to be used . . . in a computer in order to bring about a certain result" (17 U.S.C. § 101) when it decided to recognize their copyrightability. The First Circuit's interpretation flouts Congress' intention to give meaningful protection to a type of work that the statute explicitly covers. It therefore cannot be correct. As this Court has stated, it is the courts' "role to make sense rather than nonsense out of the corpus juris." *West Virginia Univ. Hosp., Inc. v. Casey*, 499 U.S. 83, 99 (1991).[66]

[65] *Id.* at 15a. The majority cited no authority for this definition. Indeed, Borland argued in the court below that the 1-2-3 menu command hierarchy was a "system," not a "method of operation," and the meaning of the latter term was not briefed or argued by either party below.

[66] *Accord, Taylor v. United States*, 495 U.S. 575, 594 (1990) (declining to follow any rule that gives statutory term a meaning that is "obsolete or inconsistent with the statute's purpose"); *United States v. Ron Pair Enter., Inc.*, 489 U.S. 235, 242 (1989) (noting that drafters' intention controls interpretation of statute whenever "the literal appli-

The fallacy of the First Circuit's reasoning is shown in its attempt to distinguish the copyrightability of the 1-2-3 program code from that of the 1-2-3 menu command hierarchy. It is true, as the court observed, that Borland need not have copied the 1-2-3 code in order "to offer the same capabilities as Lotus 1-2-3," in the sense that different program codes could have been written to achieve the same functional results. Pet. App. 16a. But that observation does *not* distinguish the copyrightability of 1-2-3's code from its menu command hierarchy. As the district court found, a program need not copy the Lotus 1-2-3 menu command hierarchy in order to offer the same capabilities as 1-2-3: different menu command hierarchies can be written to express exactly the same set of functional operations, as Borland proved with its so-called "alternative" user interfaces. *Id.* at 86a-90a.

How, then, could the First Circuit have concluded that it could apply its rule of *per se* exclusion of methods of operation to menu commands but not, with comparable effect, to program code? The answer lies in an unacknowledged shift in the First Circuit's perspective. According to the court, once Lotus 1-2-3 was written, the particular words in its menus became "essential" to its operation and, therefore, unprotectable. Pet. App. 17a. Even if this were true—and it is not[67]—it still would provide no basis for distinguishing the treatment of program code. Once a program is written, its particular statements and instructions also become "essential" to its operation. Only by considering the copyrightability of

cation of a statute will produce a result demonstrably at odds with the intentions of its drafters"). *See also* Ginsburg, *supra*, n.55 at 2569-70 (because "one should endeavor to construe statutes in a way that does not render them futile," any tension between Section 102(b) and the definition of a "computer program" must be resolved in favor of protecting "*some* ways of achieving 'a certain result' ") (emphasis in original).

67 A user who had memorized Lotus 1-2-3's menu commands and the associated keystrokes or cursor movements could operate the program without reference to its menus, just as a poem or a song can be memorized with repetition.

1-2-3's menus *after* the program was written, but its code *before*, could the First Circuit purport to find a distinction. There is simply no logical basis upon which the Court can sustain the First Circuit's interpretation of Section 102(b) without frustrating Congress' manifest intention to recognize copyright protection for computer programs.

B. The Policy Arguments Raised in the Court of Appeals Do Not Justify Depriving Computer Programs of the Statutory Protection to Which They Are Entitled

Both opinions in the court of appeals display a general discomfort with the Congressional decision to recognize copyright protection for computer programs, suggesting that the First Circuit's strained interpretation of Section 102(b) was influenced by certain policy arguments that Borland and various of its *amici* advanced concerning the alleged competitive effects of a decision in Lotus' favor. Pet. App. 21a & 23a-28a. In addition to being grossly exaggerated, most of the First Circuit's assumptions are without record support because none of these issues was tried in the district court (as discussed at pp. 15-16 above).

More fundamentally, that court appears to have ignored the plain fact that Congress resolved these policy questions in the 1976 Act.[68] As this Court previously has observed, virtually every provision in the 1976 Act was the result of a long process of negotiation and deliberation, reflecting both a carefully-crafted compromise among the interested copyright constituencies and the exercise of legislative judgment. *Stewart v. Abend*, 495 U.S. 207, 225-26 (1990); *Community for*

68 In Judge Boudin's concurring opinion, the sole citation for the proposition that courts, rather than Congress, are to determine "what may be protected and with what limitations and exceptions" (*id.* at 25a), is a single sentence in one of the lectures collected in Kaplan's *An Unhurried View of Copyright*. The lecture was delivered in 1966 and addressed, in the cited passage at page 40, the "maddeningly casual prolixity and imprecision throughout" the 1909 Act, not the as-yet-unwritten 1976 Act.

Creative Non-Violence v. Reid, 490 U.S. 730, 748 n.14 (1989); *Mills Music, Inc. v. Snyder*, 469 U.S. 153, 174 n.41 (1985).[69] This Court, accordingly, has declined to interpret provisions of the 1976 Act based on its own perception of the relevant policies. As the Court stated in *Stewart v. Abend*, 495 U.S. at 230, "it is not [the Court's] role to alter the delicate balance Congress has labored to achieve;" such "arguments are better addressed by Congress than the courts." *Id.* at 228.

For example, the concurring opinion suggests that courts should be particularly hesitant in extending copyright protection to computer programs, because to do so "can have some of the consequences of *patent* protection in limiting other people's ability to perform a task in the most efficient manner." Pet. App. at 23a (emphasis in original). Even if computer programs, standing alone, had been considered patentable subject matter at the time Congress acted—and they were not (see pp. 25-26 above)—its decision to protect computer programs under copyright law must be accepted as a deliberate and rational policy determination. The lower court's assumption that protecting the utilitarian aspects of computer programs under patent law, rather than copyright, would result in greater freedom to innovate and a lesser effect on competition is a debatable point at best.[70] Determining how

69 *See also* Miller, *supra*, n.55 at 979-85 (discussing history leading to passage of 1976 Act and its treatment of computer programs; the author was a CONTU commissioner and served as a member of its software subcommittee).

70 The numerous differences between patent and copyright law defy easy generalizations concerning which is "better" from a policy perspective. For purposes of promoting creativity, the novelty requirement for patent protection may be too high a threshold—denying *all* protection to the vast majority of original, but not truly inventive, works. If the goal is to preserve competition, patent protection may be too powerful, because a patent owner can prohibit all others from practicing the claimed invention—even innocent infringers or independent creators—unlike a copyright owner, who can demand only that subsequent authors do their own work instead of copying. *See Mazer*, 347 U.S. at 218

best to balance the countervailing factors in order to "promote the Progress of Science and useful Arts" is a task the Constitution gives to Congress, not the courts.

The First Circuit also expressed concern about the potential impact upon competition if the owners of industry-leading programs were permitted to enjoy the full protection of copyright law. Pet. App. 21a & 26a-28a. This Court, however, has rejected the notion that a copyrighted work should receive lesser rights simply due to its public importance. *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 539, 557-59 (1985). The assertion that a software product forfeits some portion of the protection to which it should be entitled because it has become—for the moment—the most popular, is "fundamentally at odds with the scheme of copyright protection." *Id.* at 559. Moreover, because these issues were not tried in the district court, there was nothing in the record to permit an accurate or measured evaluation of the legitimacy of this concern.

The concern is overstated in all events. In a competitive market, in which consumers are free to choose among substitutable—if differently expressed—offerings, the ability of any one software vendor to force unwilling adherence to a proprietary standard may prove short-lived, if not illusory.[71] When, on the other hand, the competitive marketplace has broken down, the appropriate answer is to enforce the antitrust laws, not to misinterpret copyright law.[72] Moreover,

("Absent copying, there can be no infringement of copyright."). And, although a patent confers upon its owner a shorter period of exclusivity, the difference has little practical effect in the fast-moving software industry, in which even seventeen or twenty years is more than long enough to span many product life cycles.

71 *See* Huber, *supra*, n.19 at 104 (suggesting that any decision by Lotus to overprotect its copyrights could eventually prove self-defeating, in light of consumer preference for non-proprietary standards).

72 *See, e.g., Eastman Kodak Co. v. Image Technical Services, Inc.*, 504 U.S. 451, 479 n.29 (1992) (intellectual property protection does not create immunity from antitrust laws). The doctrine of "copyright misuse"

Section 102(b), the idea/expression dichotomy, and the merger doctrine already address the First Circuit's concern. If, as in this case, many ways exist to express the same program functionality, then the market is not deprived of products performing those functions when copyright protection is accorded to the separable expression embodied in one of those ways.[73] Because this principle—which the First Circuit ignored—is embedded in the statute, there is no need or reason to create new judicial limitations that Congress never intended.

In addition, the "fair use" doctrine, under appropriate circumstances, may guard against any undue adverse effect of copyright protection on competition. Like the idea/expression distinction itself, the fair use doctrine allows some opportunity for copying of another's work to fulfill the purpose of promoting "the Progress of Science and useful Arts." *Campbell v. Acuff-Rose Music, Inc.*, 114 S. Ct. 1164, 1169 & n.5 (1994).[74] That Borland failed to make out a "fair use" defense

may also prevent a copyright owner from attempting to use its copyrights to prevent others from developing non-infringing but competitive products, and thereby obtaining excessive exclusionary power. *Lasercomb America, Inc. v. Reynolds*, 911 F.2d 970, 978-79 (4th Cir. 1990).

[73] *Cf. Two Pesos, Inc. v. Taco Cabana, Inc.*, 112 S. Ct. 2753, 2760-61 (1992) (holding that insistence upon rule that only "non-functional, distinctive trade dress is protected under § 43(a)" of the Lanham Act, 15 U.S.C. § 1125(a), which requires that a design not be "one of a limited number of equally efficient options available to competitors," is sufficient "to assure that competition will not be stifled by the exhaustion of a limited number of trade dresses").

[74] *See Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1527 (9th Cir. 1992) (finding fair use when competitor copied computer programs as only way to discover their unprotected elements, but did not include the copied expression in competing products); *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 972 (9th Cir. 1992) (finding fair use when limited portion of computer program was copied in order to provide product that worked with and complemented, but did not supersede, the original), *cert. denied*, 113 S. Ct. 1582 (1993).

in this case—or even to assert one in a timely fashion[75]—does not justify the grafting of a new, judicially-created "privilege" onto the copyright statute. Pet. App. 28a.

Bluntly put, Congress did not provide courts with a blank slate to write the rules of protection for computer software. The members of the panel below were not at liberty to negate Congress' decision solely because it did not comport with their own assessment of the various policy considerations involved. In 1976, and again in 1980, Congress spoke: expression in computer programs is entitled to copyright protection under the same principles pertaining to more traditional forms of expression, despite the fact that they are complex, utilitarian, and functional works. A large and vibrant national industry has grown in reliance on these property rights as its lifeblood. The First Circuit, based on little more than its own subjective impressions and with only the vaguest understanding of the potential consequences of its decision, ventured to supplant Congress' judgment with its own. This was an error that, we respectfully submit, must be reversed.

[75] As discussed at pp. 3-4 above, Borland only belatedly asserted fair use as a defense in this case, and then failed to prove that its conduct did not, and would not, have a damaging effect on the market for Lotus 1-2-3. *See Campbell*, 114 S. Ct. at 1176 ("Since fair use is an affirmative defense, its proponent would have difficulty carrying the burden of demonstrating fair use without favorable evidence about relevant markets.") (footnote omitted).

CONCLUSION

The judgment of the First Circuit should be reversed and this case remanded for further proceedings.

Respectfully submitted,

NEAL D. GOLDMAN
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

DONALD J. ROSENBERG
INTERNATIONAL BUSINESS MACHINES CORPORATION
44 South Broadway
White Plains, New York 10601
(914) 288-4010

HENRY B. GUTMAN
Counsel Of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

November 9, 1995

(18)

No. 94-2003

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

In The

Supreme Court of the United States

October Term, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL, INC.,

Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

BRIEF FOR RESPONDENT

GARY L. REBACK
(Counsel of Record)
MICHAEL BARCLAY
SUSAN A. CREIGHTON
COLLEEN BAL
LISA A. DAVIS
WILSON, SONSINI, GOODRICH & ROSATI
PROFESSIONAL CORPORATION
650 Page Mill Road
Palo Alto, California 94304
Telephone: (415) 493-9300

ROBERT H. KOHN
BORLAND INTERNATIONAL, INC.
100 Borland Way
Scotts Valley, California 95066
Telephone: (408) 431-1801

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY & GELHAAR
One Post Office Square
33rd Floor
Boston, Massachusetts 02109
Telephone: (617) 542-7400

Attorneys for Respondent

December 1995

COCKLE LAW BRIEF PRINTING CO., (800) 225-6964
OR CALL COLLECT (402) 342-2831

BEST AVAILABLE COPY

66 pp

RULE 29.6 STATEMENT

Respondent Borland International, Inc. has no parent corporation or subsidiaries that are not wholly owned, except for certain foreign subsidiaries in which a minimal amount of shares (fewer than 1%), which are not publicly traded, are held by foreign nationals in accordance with local law.

TABLE OF CONTENTS

	Page
RULE 29.6 STATEMENT	i
TABLE OF CONTENTS	ii
TABLE OF AUTHORITIES	iv
CONSTITUTIONAL PROVISIONS INVOLVED	1
STATEMENT OF THE CASE	1
A. Development of 1-2-3	2
1. User Interface	2
2. Words and Order	4
(a) Words	4
(b) Structure	4
(c) Code	5
(d) Description	5
(e) Written materials	7
3. Success of 1-2-3	8
B. The Development of the Borland Products	10
C. Market Effects	13
D. Procedural History of the Case	17
INTRODUCTION AND SUMMARY OF ARGUMENT	19
ARGUMENT	22
I. COPYRIGHT PROTECTION MAY NOT BE USED TO AVOID THE STATUTORY REQUIREMENTS OF THE PATENT LAWS	22

TABLE OF CONTENTS – Continued

	Page
A. The Patent Laws Carefully Limit the Extent to Which the "Useful Arts" May be Subject to Private Monopoly	24
B. *Baker v. Selden* and its Progeny Have Rejected the Use of Copyright to Circumvent the Requirements of the Patent Statute	26
1. The Court's *Baker v. Selden* Decision	27
2. Cases following *Baker v. Selden*	29
C. Section 102(b) Incorporates the Limitation on Copyright Protection For Utilitarian Functions	31
1. The Terms in Section 102(b) are Drawn From the Patent Law	31
2. A "Computer Program" is not a "Process" or "Method of Operation."	35
II. LOTUS' MENU COMMAND HIERARCHY IS UNPROTECTED UNDER SECTION 102(b) OF THE COPYRIGHT ACT	38
A. The Lotus Menu Command Hierarchy is Not a "Computer Program" Under Section 101	39
B. The Menu Command Hierarchy is an Unprotected Element of the Lotus Users' Manual and the Lotus Screen Display	41
1. The Lotus Reference Manual	42
2. The Lotus 1-2-3 Screen Display	43
C. Even if Part of the Computer Program for Purposes of Section 102(a), the Menu Command Hierarchy is an Unprotected Method of Operation under Section 102(b)	46

TABLE OF CONTENTS – Continued

Page

APPENDIX

Lotus 1-2-3 Screen Display A-1

Borland's Quattro Pro Screen Display A-2

Lotus Command Tree A-3

Lotus 1-2-3 Screen Display (showing long prompt) A-4

Microsoft's Excel Screen Display A-5

TABLE OF AUTHORITIES

Page(s)

CASES

Affiliated Enters. v. Gruber, 86 F.2d 958 (1st Cir. 1936) 29n, 33

Amberg File & Index Co. v. Shea Smith & Co., 82 F. 314 (7th Cir. 1897) 33

Application of Drummond, 302 F.2d 761 (C.C.P.A. 1962) 34n

Application of Hortman, 264 F.2d 911 (Cust. & Pat. App. 1959) 33

Application of Horvath, 211 F.2d 604 (C.C.P.A. 1954) ... 32n

Application of Moreton, 288 F.2d 708 (C.C.P.A. 1961) ... 32n

Application of Orsini, 158 F.2d 286 (C.C.P.A. 1946) ... 34n

Application of Schutt, 210 F.2d 293 (C.C.P.A. 1954) ... 32n

Baker v. Selden, 101 U.S. 99, 25 L.Ed. 841 (1879) ..... *passim*

Beedle v. Bennett, 122 U.S. 71 (1887) 26n

Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989) 20, 25, 28, 38n

Brief English Sys., Inc. v. Owen, 48 F.2d 555 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931) 29n, 33

Campbell v. Acuff-Rose Music, Inc., 114 S.Ct. 1164 (1994) 18n

Capitol Records v. Mercury Records Corp., 221 F.2d 657 (2d Cir. 1955) 23n

Coakwell v. United States, 372 F.2d 508 (Ct. Cl. 1967) 26n

Crume v. Pacific Mut. Life Ins. Co., 140 F.2d 182 (7th Cir.), *cert. denied*, 322 U.S. 755 (1944) 29n

TABLE OF AUTHORITIES – Continued

Page(s)

Davis v. Michigan Dept. of Treasury, 489 U.S. 803 (1989) 33

Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518 (1972) 23

Diamond v. Chakrabarty, 447 U.S. 303 (1980) 23n, 32

Diamond v. Diehr, 450 U.S. 175 (1981) 34

Expanded Metal Co. v. Bradford, 214 U.S. 366, 53 L.Ed. 1034 (1909) 32n

Feist Publications, Inc. v. Rural Telephone Svc. Co., 499 U.S. 340 (1991) 20, 37n

Fogerty v. Fantasy, Inc., 114 S.Ct. 1023 (1994) 2, 20, 22, 23, 38, 45

Gottschalk v. Benson, 409 U.S. 63 (1972) 34n

Graham v. John Deere Co., 383 U.S. 1 (1966)... 23n, 24, 38n

Graver Tank & Mfg. Co., Inc. v. Linde Air Prods. Co., 336 U.S. 271 (1949) 34n

Griggs v. Perrin, 49 F. 15 (C.C.N.D.N.Y. 1892)..... 33

Gustafson v. Alloyd Co., Inc., 115 S. Ct. 1061 (1995) 31

In re Hayes Microcomputer Prods. Patent Litig., 982 F.2d 1527 (Fed. Cir. 1992) 26

Kouf v. Walt Disney Pictures & Television, 16 F.3d 1042 (9th Cir. 1994)..... 35n

LeRoy v. Tatham, 55 U.S. 156 (1852) 33

Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992)..... 2n, 17, 35

Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992)..... *passim*

TABLE OF AUTHORITIES – Continued

Page(s)

Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 202 (D. Mass. 1993)..... 2n, 4n, 8n, 9, 17

Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993)..... *passim*

Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) ..... *passim*

Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990)..... 2, 3, 8, 8n, 10, 19n, 43

Mazer v. Stein, 347 U.S. 201 (1954) 30

Morrissey v. Proctor & Gamble Co., 379 F.2d 675 (1st Cir. 1967) 4n

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)..... 35n

Raffold Process Corp. v. Castanea Paper Co., 98 F.2d 355 (3d Cir.), *cert. denied*, 305 U.S. 635 (1938)..... 34

Russell Motor Car Co. v. United States, 261 U.S. 514 (1923) 32

Saranac Automatic Mach. Corp. v. Wirebounds Patents Co., 282 U.S. 704 (1931)..... 34n

Sony Corp. of America v. Universal City Studios, Inc., 464 U.S. 417 (1984)..... 22, 23n, 38, 49

Taylor Instrument Cos. v. Fawley-Brost Co., 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944) 29, 30

Twentieth Century Music Corp. v. Aiken, 422 U.S. 151 (1975) 38, 38n, 49

TABLE OF AUTHORITIES – Continued

Page(s)

U.S. Indus. Chems., Inc. v. Carbide & Carbon Chems. Corp., 315 U.S. 668 (1942) 34n

Voightmann v. Perkinson, 138 F. 56 (7th Cir. 1905) 33

STATUTES

17 U.S.C. § 101 (1995) 39

17 U.S.C. § 102(a) (1995)..... 21, 22, 23n, 46, 50

17 U.S.C. § 102(b) (1995) ..... *passim*

17 U.S.C. § 106(1) (1995) 26

35 U.S.C. § 100(b) (1995) 32

35 U.S.C. § 101 (1995) 32

35 U.S.C. § 102 (1995) 19

35 U.S.C. § 103 (1995) 19

35 U.S.C. § 112 (1995) 19, 35

35 U.S.C. § 131 (1995) 19

35 U.S.C. § 154 (1995) 26

Act of Feb. 21, 1793, § 1, 1 Stat. 319 ..... 20n, 23n

LEGISLATIVE MATERIALS

H.R. REP. NO. 1923, 82d Cong., 2d Sess. (1952) 23n, 24n

H.R. REP. NO. 1476, 94th Cong., 2d Sess. (1976).. 23n, 35

S. REP. NO. 338, 24th Cong., 1st Sess. (1836)..... 45n

S. REP. NO. 983, 93d Cong., 2d Sess. (1974)..... 33

TABLE OF AUTHORITIES – Continued

Page(s)

Copyright Law Revision: Hearings Before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on The Judiciary, 90th Cong., 1st Sess. (1967) 36

Copyright Law Revision: Hearings Before the Subcomm. on Courts, Civil Liberties, and the Admin. of Justice of the House Comm. on the Judiciary, 94th Cong., 1st Sess. (1975) 34n

CONSTITUTION

U.S. CONST. art. I, § 8, cl. 8 ..... 1, 19, 23n, 33

MISCELLANEOUS

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1979) 37, 38, 39, 39n, 41n, 47

Benjamin Kaplan, AN UNHURRIED VIEW OF COPYRIGHT 64 (1967) 28, 29

IBM DICTIONARY OF COMPUTING (10th ed. 1993) 40n

1 Paul Goldstein, COPYRIGHT § 2.15.3.1 (2d ed. 1996) ... 41n

1 KAMINSTEIN LEGISLATIVE HISTORY PROJECT (1981) 36

Kenneth A. Liebman, et al., *Back to Basics: A Critique of the Emerging Judicial Analysis of the Outer Limits of Computer Program "Expression,"* 2 COMPUTER LAW., (Dec. 1985)..... 46n

MICROSOFT PRESS COMPUTER DICTIONARY (2d ed. 1994) ... 40n

TABLE OF AUTHORITIES – Continued

Page(s)

U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (Washington, D.C.: Government Printing Office, May 1992) 48n

WEBSTER'S NINTH NEW COLLEGIATE DICTIONARY (9th ed. 1988) 33, 34, 55n

William M. Landes and Richard A. Posner, *An Economic Analysis of Copyright Law*, 18 J. OF LEGAL STUD. (1989) 37n

Borland International, Inc. ("Borland") respectfully seeks affirmance of the judgment of the United States Court of Appeals for the First Circuit in this case.

CONSTITUTIONAL PROVISIONS INVOLVED

Lotus Development Corporation ("Lotus") does not cite in full the applicable Constitutional provision involved. Brief for the Petitioner ("Lotus Br.") at 1, 20. Article 1, Section 8, Clause 8 of the U.S. Constitution provides (portions omitted by Lotus italicized):

> "Congress shall have the power . . . To promote the progress of science and useful arts, by securing for limited times to authors *and inventors* the exclusive right to their *respective* writings *and discoveries*."

STATEMENT OF THE CASE

There is good reason why this computer software copyright case has generated such widespread attention among the computer industry, the academic community and the legal profession. In the usual software copyright case, the defendant is alleged to have copied either the text ("code") or structure of plaintiff's computer program, or the way the program looks on the computer screen when it is executing. No such copying occurred here. Lotus does not even allege any copying of its code, and the district court held that the programs at issue look different, a ruling Lotus did not attempt to appeal.

The sole basis for Lotus' claim is that Borland copied menus of command words that are displayed on the computer screen to permit the user to invoke the functionality of the program. Lotus does not allege that Borland copied the textual explanations of the command words – only the words themselves and their order. The words (such as "COPY" and "PRINT") in Lotus' menus are like the labels on switches and knobs. In number, arrangement and function, they are analogous to the switches and attendant labels that are arranged and grouped in the cockpit of a commercial jet aircraft. Indeed, the district court characterized Lotus' menu hierarchy as "the selection and arrangement of executable operations."

Lotus' menu command hierarchy may well constitute patentable subject matter. But Lotus bases its claim on the law of copyright, not patent. Lotus' argument ignores the important policies and procedures of the patent law, reads important limiting language out of the copyright statute, and refuses even to acknowledge that its proposed extension of the scope of copyright must "ultimately serve the public good," as this Court has so often counseled. *See, e.g., Fogerty v. Fantasy, Inc.*, 114 S.Ct. 1023, 1029 (1994). The First Circuit Court of Appeals unanimously rejected Lotus' approach. Borland respectfully requests that this Court do so as well.

A. Development of 1-2-3

1. User Interface

The record in this case indicates that the code and some of the functionality of the 1-2-3 program were original to Lotus, but virtually every element of the program's user interface, including the command words and segments of the hierarchy, was present in products developed by other companies prior to 1-2-3. VisiCalc, the first commercial electronic spreadsheet, was described by the district court as "a revolutionary advance in the field of programming." *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 65 (D. Mass. 1990) ("*Paperback*"), Pet.App. 230a.[1] The extent to which Lotus appropriated aspects of VisiCalc was hotly contested in the *Paperback* case. *See id.* at 83, Pet.App. 265a. The district court conceded that Lotus used a number of elements from VisiCalc, including the spreadsheet metaphor and the designations of various keys to perform specific functions, *see id.* at 66, Pet.App. 231a-232a, but the court found each of these features uncopyrightable. The court concluded that the 1-2-3 menu structure

[1] Each of the four decisions of the District Court in *Lotus Dev. Corp. v. Borland Int'l, Inc.* relies upon *Paperback*. Those opinions are reported at 788 F. Supp. 78 (D. Mass. 1992) ("*Borland I*"), 799 F. Supp. 203 (D. Mass. 1992) ("*Borland II*"), 831 F. Supp. 202 (D. Mass. 1993) ("*Borland III*"), and 831 F. Supp. 223 (D. Mass. 1993) ("*Borland IV*"). The decision of the Court of Appeals appears at 49 F.3d 807 (1st Cir. 1995) ("*Lotus v. Borland*").

"taken as a whole" was "original and non-obvious" and, hence, the 1-2-3 user interface was copyrightable. *Id.* at 68, Pet.App. 234a-235a.

The 1-2-3 user interface presents command choices through a "two-line moving cursor." The top line is the set of available command options, and the second line is the "long prompt," or explanatory message, for the command on which the cursor is set in the top line.[2] *Id.* at 64, Pet.App. 227a. Commands are selected either by (a novice) using the cursor keys to highlight the desired command, then pressing "enter," or by (a more advanced user) typing the first letter of the command on the keyboard, in the same manner as a touch typing system. *See, e.g.*, Pet.App. 288a (Kapor Aff.); *Paperback* at 64, Pet.App. 228a. Commands may also be selected when they are spoken into a computer equipped with voice recognition software. *See Lotus v. Borland* at 816, Pet.App. 17a.

Comparing 1-2-3's full-word menu presentation with the list of letters displayed by VisiCalc, Lotus asserts that its menu command hierarchy represented "a major advance in interface design." Lotus Br. at 11. The presentation of full words, however, like most of the rest of 1-2-3's user interface, was not original to Lotus – full word display was taken from a different product, VisiTrend/Plot, which also predated 1-2-3. VisiTrend/Plot was a graphics and statistical analysis program that 1-2-3's principal developer, Mitchell Kapor, worked on for a previous employer, Personal Software, Inc. Pet.App. 282a-283a (Kapor Aff.). Kapor's own affidavit, submitted by Lotus in this case, admits that the two line moving cursor and its "main elements" came from the earlier VisiPlot product. *Id.* at 286a-287a. Indeed, the record in this case demonstrates that virtually all of the user interface features associated by the district court in *Paperback* with 1-2-3 were taken initially from the VisiPlot product. *See* JA 311 (Buechele Decl.). These include features such as "full word command names," and "command long prompts following highlighting on status line."[3]

[2] A picture of a 1-2-3 screen with the menu commands and "long prompt" identified is found at A-4 in the Appendix to this brief and at JA 981.

[3] Other such features include a "highlighted, moving cursor," "command selection based on highlighting, with command activation based on subsequent

2. *Words and Order*

(a) Words. It is equally undisputed that the vast majority of command words and functions of 1-2-3 were not original to that product. Many of the 1-2-3 menu commands and functions were present in VisiCalc. The early drafts of the menu command hierarchy of 1-2-3 were in substantial part identical to that of VisiCalc,[4] and commands as well as functionality were incorporated into 1-2-3 from Context MBA (an applications program) and BASIC (a programming language).[5]

It is certainly no surprise that 1-2-3 would use the same command words as pre-existing products because, according to Jonathan Sachs, one of the 1-2-3 developers, "every command was chosen because it suggested to some measure what the command did." JA 750 (depo.). The Kapor affidavit is to the same effect: words were chosen to "intelligently convey to the user the purpose of each command and its underlying functionality." Pet.App. 291a.[6]

(b) Structure. Lotus does not base its claim of infringement on Borland's use of the same command words, since those words are common to spreadsheet products.[7] Lotus' claim of infringement therefore devolves to the question of the copyrightability of the

---

<Enter> key," "command activation based on abbreviated command name," and "ordering of commands by anticipated frequency of use." JA 311 (Buechele Decl.).

[4] Lotus doc. 046195 LP; *compare* Lotus doc. KP00861 *with Paperback* at 67, Pet.App. 234a. The similarity was of both words and order. The menu command hierarchy was reordered shortly before commercial release to make it more functional. *Compare* Lotus doc. 046559, 582-608 (command structure as of Sept. 7, 1982) *with* Lotus doc. 046685-91 (final structure). The cited Lotus documents are attached to the Konstantaras Decl. (Dkt. 230), Ex. A, H, I [Dkt. 241, 3:14-3:18; Dkt. 242, 4:22-5:24].

[5] Lotus docs. KP02306; KP01103; KP01082; KP01085 (Dkt. 230), Ex. B, C [Dkt. 241, 3.14-3:18; Dkt. 242, 4:22-5:24].

[6] In copyright terms, each menu command and its underlying function "merged." *See, e.g., Morrissey v. Proctor & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967).

[7] For example, Quattro Pro's "native" interface, against which Lotus professes no claim, uses basically the same command set in a different order. Lotus makes no claim to the Quattro products' native interfaces. *Borland III* at 211, Pet.App. 82a.

order or *structure* of these common commands, but not the *words* themselves. Lotus Br. at 6.

According to Lotus' affidavits, the commands were organized "hierarchically," in the manner depicted in the "menu tree," so that "the selection of one command option from the first level menu could lead in turn to another array of command options on a second level menu (or 'submenu'), and so on."[8] Pet.App. 287a (Kapor Aff.). This "menu tree" is merely a depiction setting out the organized hierarchical set of alternative steps that an individual may take to manipulate and perform calculations and other operations on the data contained in the cells of the spreadsheet grid. As the program executes, a few commands appear in the top portion of the screen in an order determined by the functional result the user is seeking. The hierarchical arrangement was chosen to enable "the novice user to browse through the menu levels, in order to view the valid sequences of available options (and their corresponding explanations) and to map out a plan for performing a particular task." Pet.App. 287a (Kapor Aff.).

(c) Code. Lotus' brief implies that 1-2-3's developers first prepared their product's code and then affixed labels to the various functions. *See* Lotus Br. at 9. In fact, however, the record submitted by Lotus in the trial court reveals a far more iterative process, during which changes in the hierarchy were implemented (or "expressed") in code. Kapor started borrowing user interface features from VisiPlot in December of 1980 and began to apply these ideas to his new product in mid-1981. Pet.App. 285a (Aff.). "Typically, Kapor suggested a feature; we [Kapor and Sachs] discussed it; and [Sachs] attempted to implement it in the program's source code." JA 538 (Sachs Aff.). In 1982, Kapor made the final decisions as to "what words would be used and where on the menu tree they would be located. [Sachs] then modified the source code to reflect [Kapor's] changes." *Id.* at JA 539.

(d) Description. Lotus describes the purpose of the menu command hierarchy in terms that differ markedly from the record in the district court. Lotus asserts that a "proper understanding of

---

[8] A chart from the trial exhibit depicting a portion of this "tree" is reproduced in the Appendix to this brief at A-3. JA 918. The entire menu tree is depicted at Ex. C to Flesher Decl. (Dkt. 81) [Dkt. 164, JA 78-79].

the nature and purpose of the 1-2-3 menu command hierarchy is critical to an appropriate resolution of this case." Lotus Br. at 5. But Lotus omits any mention of the other elements of 1-2-3 that explain and describe the hierarchy, leading the reader to conclude that the hierarchy must, of necessity, fill the role of providing explanation, as well as invoking functionality. Without any citation to the record, Lotus argues, for example, that the menu words are "a form of structured dialog between 1-2-3's authors and users." *Id.* at 6. For purposes of this appeal, the words supposedly are "simply words of text" that "provide information" in "plain English" like "the pages of an instruction manual." *Id.*

In fact, the Lotus commands are no more "an instruction manual" than are the "■" or "►►" buttons on a videocassette recorder ("VCR") the same as the VCR instruction manual. Lotus' key witnesses, documents and experts in the district court all drew a firm line between the functional role of the menu commands in the hierarchy (the "options" for "performing a particular task") and the "explanation" of those commands provided elsewhere in the product. The affidavit of 1-2-3's principal developer, for example, described the elements of the 1-2-3 user interface in terms similar to those employed by this Court in *Baker v. Selden*. Thus, according to the Kapor affidavit, the object of the long prompts is to provide "information to the user" and "explanations," while the object of the menus is to perform "a particular task." Pet.App. 287a, 296a. *Compare* 101 U.S. 99, 105 (the object of copyrightable subject matter is "explanation," while the object of patentable subject matter is "use").

Lotus' expert Galler also distinguished between the menu choices themselves and "explanatory information" – in the form of long prompts and help screens – about the menu choices. JA 376-77, 381 (Galler Decl.). As the Galler declaration (submitted by Lotus) makes clear, the command words do not explain to the user how to use the system – they identify the functions that comprise the system and provide the means by which each function is invoked. Explanation and education are provided by the on-line help facility, product documentation, long prompts, and tutorials supplied with the system. *Id.* at 377; *see also* JA 393-94 (2nd Galler Decl.) (admitting command words do not provide sufficient

information to be a substitute for the documentation, tutorials or help screens).

(e) Written materials. The written materials that accompany the Lotus product reflect the same dichotomy between the functional words of the hierarchy and their accompanying description. According to Lotus' written materials, the menus are "options" that the user may "select."[9] A long prompt, on the other hand, is a "brief description of what the command does."[10] A more complete explanation can be found in the on-line "Help" text that the program displays on the screen: "The Help facility is like a reference manual that is always open to the right page."[11] And, finally, of course, the written documentation provided by Lotus "contains detailed information about all of the features of 1-2-3."[12]

The commands are the means by which a user invokes the functionality of the program. Each individual command and each sequence of commands is described and explained in the long prompts, the on-line help text, and the user's manual. *See* 1-2-3 Ref. at 44 (Dkt. 158). Tables illustrating the descriptions given for particular command sequences in each of these sources are set out in the record as part of Borland's motion for summary judgment in the District Court. *See* Borland's Mem. in Support of Renewed Motion for S.J. (Dkt. 168) at 55, 56.

Borland does not deny that the user's manual, on-line help text, and perhaps even the long prompts may contain copyrightable "expression." But Lotus has not accused Borland of copying such explanatory text. Lotus' claims are directed only to the menu command hierarchy which Lotus' own witnesses and documents readily distinguish from "explanation" and "description."

---

[9] Lotus 1-2-3 User's Manual (1983), Borland's Mem. in Support of Cross-Motion for S.J. ("Borland S.J. Br."), Ex. 9 ("S.J. Ex.") (Dkt. No. 142) at L084455 [Dkt. 164, JA 78-79].

[10] Lotus 1-2-3 Reference Manual, Release 2 at 28 (1st ed. 1985) ("1-2-3 Ref.") (Dkt. 158) [Dkt. 164, JA 78-79]. The First Circuit noted that "[t]he long prompts explain, as a sort of 'help text,' what the highlighted command will do if entered." *Lotus v. Borland* at 811 n.2, Pet.App. 7a.

[11] Lotus 1-2-3 Tutorial, Release 2 at 12 (1st ed. 1985) ("1-2-3 Tut.") (Dkt. 158) [Dkt. 164, JA 78-79].

[12] 1-2-3 Tut. at Preface (Dkt. 158) [Dkt. 164, JA 78-79].

Nor was the district court under any illusion that it was protecting something akin to "the pages of an instruction manual." Lotus Br. at 6. There was no misunderstanding by the district court as to "the nature and purpose" of the subject matter at issue. Given the record created by Lotus,[13] the district court forthrightly and correctly described the menu command hierarchy in starkly functional terms – as the "selection and arrangement of executable operations," which the district court held to be copyrightable subject matter. *Borland IV* at 231, Pet.App. 41a. The district court was equally unambiguous in holding that copying functionality constitutes copyright infringement: "Borland's reason for copying the menu command structure was to obtain the benefits of its functionality." *Borland II* at 209, Pet.App. 115a. Having labored to obtain exactly what it sought from the district court – protection *by copyright* over the "selection and arrangement of executable operations" – Lotus should not now be heard to suggest that it is really the "explanation" as opposed to the "system" or "method of operation" that it is trying to protect.

3. Success of 1-2-3

1-2-3 was a market success, but that success had little to do with the menu command hierarchy. As the district court observed in *Paperback*, VisiCalc, the predecessor of 1-2-3, was programmed for use on the Apple II computer, which had limited functional capabilities. *Paperback* at 65, Pet.App. 230a-231a. When the IBM PC was introduced in August of 1981, the developers of 1-2-3 "exploited this opportunity" by designing 1-2-3 to take "advantage of the IBM PC's more expansive memory and more versatile

---

[13] The district court record also reflects the fact that the 1-2-3 hierarchy was organized according to a number of functional principles: predicted frequency of use of commands, approximately seven functions on a menu level, unique first letters on each level, etc. *See, e.g., Paperback* at 67, Pet.App. 234a; Pet.App. 291a-292a (Kapor Aff.). For example, commands within a given menu level were arranged from left to right in declining order of frequency of usage, so as to minimize keystrokes. Pet.App. 291a (Kapor Aff.). The lower court found that these principles functioned merely as "guidelines" and did not fully constrain the choice of words or order *at the time Lotus 1-2-3 was first developed. Borland III* at 213, Pet.App. 85a.

screen display capabilities and keyboard." *Id.* at 66, Pet.App. 231a.[14] The menu command hierarchy was simply not a qualitatively significant part of the product *at the time of its introduction*, either from Lotus' viewpoint or the viewpoint of users. As the district court observed, citing Borland's experts, "the words Lotus selected did not matter for 1-2-3's success." *Borland III* at 213, Pet.App. 87a.[15]

The exact words and order of the Lotus menu command hierarchy were not important to the product's initial success, but they *became* vitally important to the success of later 1-2-3 versions and spreadsheets offered by Lotus' competitors because of the "macro" capability included within the first 1-2-3 release. The citations for Lotus' assertion that "both sides experts agreed" that the menus possessed "great commercial significance," Lotus Br. at 11, all refer to the period of time *after* users invested heavily in the creation of "macros." *See* JA 343-344 (Emery Decl.); JA 507 (Olson Decl.).

Like the other 1-2-3 features, macro capability was not original to 1-2-3; it was taken from pre-existing products. Pet.App. 290a (Kapor Aff.); JA 529-30 (Raburn Decl.). From the initial release of 1-2-3, the documentation for the product instructed users to create "keyboard macros." JA 530 (Raburn Decl.) & JA 535-36 (Ex. B); JA 877 (Ex. 38) [Dkt. 164, JA 78-79]. "Macros" are application programs that users and third parties write, using the

---

[14] The former Lotus executives responsible for the "launch" and initial marketing of 1-2-3 testified that the menu commands played no ascertainable role in its commercial success: "Any word would have sufficed for a command so long as the word conveyed to the user in a direct and simple way the function of the command." JA 528 (Raburn Decl.); *see* Borland S.J. Br. (Dkt. No. 141) at 134-35; JA 527 (Raburn Decl.); JA 404-06 (Goldschmitt Decl.). An internal Lotus memorandum prepared shortly before 1-2-3's introduction did not identify the command hierarchy as a feature that would cause 1-2-3 to be a success. JA 405-06 (Goldschmitt Decl.); S.J. Ex. 31 (Dkt. 142) at KP02216-KP02224 (Product Positioning Summary, Sept. 7, 1982) [Dkt. 164, JA 78-79].

[15] Nor did Lotus' expert dispute the point. Rather, he deemed the question of whether other words could have been changed at the time of 1-2-3's development to be "irrelevant." JA 343 (Emery Decl.). The only finding the district court made with respect to the menu words was that the menus used were a "more than trivial" portion of the program. *Borland II* at 219, Pet.App.135a.

words and order of the 1-2-3 menu command hierarchy as a "command language." Pet.App. 290a (Kapor Aff.). A simple macro is a stored set of instructions that can be invoked in a single keystroke, *see Paperback* at 64, Pet.App. 228a, but more complicated macros, consisting of thousands of lines of code, perform sophisticated applications, *see Borland IV* at 227, Pet.App. 32a; JA 508 (Olson Decl.) (example of more complex macro).

A macro represents a set of steps that must be performed in a designated way – that is, according to the words and order of the 1-2-3 hierarchy. Synonyms for the words and/or an alternative order simply will not work. *See Borland II* at 213-14, Pet.App. 122a-125a. As the district court observed, "the exact hierarchy – or structure, sequence and organization – of the menu system is a fundamental part of the functionality of the macros." *Paperback* at 65, Pet.App. 229a.[16]

B. The Development of the Borland Products.

Borland's objective was to design spreadsheets that were far superior to existing spreadsheet products, not to produce a Lotus look-alike or "clone." *See, e.g.*, JA 575-76 (Bosworth Depo.). It took Borland's team of engineers nearly three years to produce Borland's first spreadsheet, Quattro. *Id.* at 541-43. As the Court of Appeals observed, Pet.App. 4a, Quattro included enormous innovations over competing spreadsheet products, including Lotus 1-2-3.[17]

---

[16] Obviously, a user cannot write a 1-2-3 macro – *i.e.*, use the words and order of the 1-2-3 hierarchy as a programming language – unless those words and their order are displayed to the user. Similarly, if the user is writing the macro for use by others, those users cannot rewrite, debug (*i.e.*, correct) or modify the macro (nor can the author) without visual access to the 1-2-3 hierarchy. *See, e.g.*, JA 763 (Warfield Depo.). While some macros can be executed (*i.e.*, run by the program) without the command words being displayed, the district court found that macros which require input from the user to complete their execution – "interactive macros" – require display of the exact words and order of the hierarchy to enable execution. *See Borland IV* at 227, Pet.App. 32a.

[17] These features, affecting the program's functionality, user interface and menu command hierarchy, were set forth in detail in Borland's S.J. Br. (Dkt. 141) at 52, 53 and specifically cited to the Court of Appeals. *See* Borland 1st Cir. Br. at 16.

Quattro Pro was first introduced in 1989 and won every major award for spreadsheet excellence given in the software industry. *See* Borland's S.J. Br. (Dkt. 141) at 2. Lotus' assertion that the record lacks proof as to the "inherent product superiority" of Borland's programs, Lotus Br. at 15, is simply incorrect. Borland's proof in both the district court and the Court of Appeals came from Lotus' own documents. For example, Quattro Pro invariably ranked substantially higher than 1-2-3 in head-to-head reviews and user comparisons, including those conducted by Lotus.[18] As early as 1988, a Lotus internal study showed that 1-2-3 users no longer considered the 1-2-3 user interface as the best user interface. JA 832. Lotus spreadsheets were viewed as far less technologically advanced than those of either Borland or Microsoft. JA 882-883. And Lotus' own "Quattro Pro Displacement Study" stated that two-thirds of spreadsheet users rated Quattro or Quattro Pro as the best spreadsheet on the market. JA 864.[19]

The Borland products were written in wholly original code. Lotus' brief seems to imply some similarity of code by stating that the menu words are "spelled out, in text, in the program code," Lotus Br. at 7, but any such suggestion is wholly without support. At no time in the proceedings below did Lotus ever claim code similarity or that Borland copied Lotus' computer code, nor was the code of any product ever submitted as evidence at any point in the proceedings below.[20] In fact, the district court expressly held

---

[18] JA 821-826; Borland S.J. Br. (Dkt. 141) at 2; Borland 1st Cir. Br. at 17.

[19] S.J. Ex. 14, 27, 40 (Dkt. 142) [Dkt. 164, JA79]. Nor did the Court of Appeals fail to "realize that there were newer versions of Lotus' products" in the record. Lotus Br. at 15 n. 25. Rather, as Borland argued to the Court of Appeals, Borland 1st Cir. Br. at 2, the Complaint charges infringement of only Releases 1.0, 1A and 2.0 of the Lotus product, and subsequent versions were placed in the record over Borland's objections. JA 15-16 (Complt.). The record demonstrates that Lotus copied into these subsequent versions of its own product features unique to Quattro at the time of Quattro's release. These features are described in detail at Borland S.J. Br. (Dkt. 141) at 53.

[20] Lotus' record cites refer only to a portion of a Borland data file which was placed in the record to show the existence of an abbreviated version of the Lotus menu tree. While Lotus asserts that the menu words were "hidden inside" the Borland program, Lotus Br. at 15, it is uncontroverted that "the menus themselves are divorced from being any part of any executable code." JA 792 (Warfield Depo.).

that code was not at issue in the case. Trial Tr. of Apr. 1, 1993, (Dkt. 404) JA 299-300.

The user interface of Quattro Pro is different in every respect (save some of the command words) from that of the 1-2-3 versions at issue in this case. Indeed, the differences in user interface are so striking that, as the First Circuit noted, Lotus did not even cross-appeal from the district court's finding that the Borland interface "looks substantially different from the 1-2-3 user interface." *See Lotus v. Borland* at 18, Pet.App. 9a-10a; *Borland II* at 220, Pet.App. 137a.[21]

The Quattro products were shipped with a number of different menu command hierarchies. In both products, a completely original menu command hierarchy is the "native" or default mode that is automatically presented to the user. The user is required to install any of the alternative hierarchies, including the 1-2-3 compatible hierarchy. Although Lotus suggests that users would readily employ the compatible menus to manipulate the spreadsheet, Lotus Br. at 13, Lotus' own documents belie any such suggestion. Lotus' 1990 study, for example, confirmed that Quattro users use the native menus for spreadsheet functionality, reserving the 1-2-3 compatible modes to run macros, which "continues to be important, because users frequently exchange files with 1-2-3 users." JA 871.[22] S.J. Ex. 27 (Dkt. 142) [Dkt. 164, JA 78-79].

Similarly, Lotus made no showing whatsoever as to any relationship between its menus and its own code, and it would be technically incorrect to infer any specific relationship between the code and the commands of the screen. *See, e.g.*, JA 450 (Liddle Decl.).

[21] Figures 1 and 2 in the Appendix to this Brief (pp. A-1, A-2) compare the Lotus and Borland screen displays, as the programs look when engaged in the same operation. JA 979-80. *See also* Borland's S.J. Demo Video (Dkt. 90) [Dkt. 164, JA 78-79], copies of which are lodged in the clerk's office.

[22] The menus that provide 123-compatibility in Borland's products are not merely a copy of the 1-2-3 command hierarchy. Rather, the 123-compatible menus also contain all of the functionality of the command sets from the native mode. As a result, the 1-2-3 menu command hierarchy sequences, designed by Lotus for a different menu presentation and function set, is clumsy and dysfunctional when used with the Quattro and Quattro Pro menu presentations. JA 565-566 (Bosworth Depo.); JA 520-522 (Olsqn Decl.). According to a study conducted by Burke Marketing Research, one of the largest and most respected market research

As the Court of Appeals explained, Pet.App. 4a-5a, Borland incorporated 1-2-3 hierarchy command sequences into the visual display of its compatible mode for two reasons: first, to enable spreadsheet users who were already familiar with Lotus 1-2-3 "to switch to the Borland programs without having to learn new commands," and second, because there was no other way to achieve complete macro compatibility – *i.e.*, to enable users to run, modify and debug macros originally created for use in 1-2-3.[23] The designers of the Borland products neither contemplated nor desired that the 1-2-3 commands be used routinely in place of the native command hierarchy.[24]

C. Market Effects

Lotus asserts that the anticompetitive effects flowing from its assertion of copyright protection are not sufficiently documented in the record to support the First Circuit's decision. *See* Lotus Br. at 15-16, 45-47. Specifically, Lotus claims that Judge Boudin's discussion of both user "lock in" and appropriation by Lotus of end user investment in the hierarchy are "without record support" and "derived from a non-existent record." *Id.* at 45, 16. This argument is preposterous.

In its initial summary judgment memorandum, Lotus argued that its user interface, specifically its menu command hierarchy, was the most important part of its product. JA 366 (Galler Decl.). Borland countered that Lotus' particular words and order were not

---

organizations in the country, only about 12% of Quattro and Quattro Pro users use the 123-compatible menus. Indeed, two-thirds of those who have *ever* used the 123-compatible menus cite the running of 1-2-3 macros as a reason for doing so, and 35% use the compatible menus *only* when someone else gave them a worksheet with 1-2-3 macros. JA 308 (Boyd Decl.).

[23] JA 656-57, 658-59 (Kahn Depo.); JA 553-555 (Bosworth Depo.). Lotus incorrectly asserts that Borland's executives "could not swear" that macro compatibility "was *necessary* to allow Borland to offer a commercially viable product," Lotus Br. at 15 (emphasis in original). As Borland's president put it, "macro compatibility was the key to being able to have a product that could have any chance in the marketplace." JA 657 (Kahn Depo.).

[24] JA 656-57 (Kahn Depo.); JA 566 (Bosworth Depo.).

commercially significant at the time of the first release of Lotus' product, *see, e.g.*, JA 454 (Liddle Decl.); JA 507 (Olson Decl.), and Lotus responded that the precise words and order of its hierarchy became extremely valuable through the investments of users and other third-parties, *see, e.g.*, JA 600-01 (Emery Depo.) – a conclusion with which Borland agrees.

As a consequence of Lotus' strategy, Borland was able to compile an exhaustive record of market effects, based solely on the testimony of Lotus' executives and expert witnesses and on the text of Lotus' documents. Judge Boudin's comparison of the Lotus menus to the QWERTY keyboard, Pet.App. 24a, 26a, for example, is no more than an echo of the testimony of Lotus senior executive Frank Ingari, who argued that the "fingertip knowledge" of "millions" of 1-2-3 users, comparable to how "you and I might type on a Qwerty keyboard," provides "an extremely strong incentive for users to stay with Lotus." JA 649-50. Lotus expert James Emery agreed, arguing that users "would be unwilling to change" from the 1-2-3 menus because that would not only involve "foregoing" the "learning that they invested in it, but a whole set of auxiliary advantages of third [party] products, trained user population, textbooks, et cetera." JA 601 (depo.).

Indeed, the most compelling testimony concerning market effects came from Emery, who, in describing how the Lotus menus came to be valuable, delivered a textbook definition of what economists call a "network effect":

> There tends to be a whole structure that grows up around the successful product and we get a positive feedback mechanism, that the value of a product increases greater value, which in turn, further increases the value through all these ways.

JA 600-01 (depo.). Emery therefore concluded that the value of the 1-2-3 menus "lies precisely in the familiarity that millions of 1-2-3 users have acquired with them." JA 343-44 (decl.); JA 600 (depo.).

The Lotus documents tell an equally compelling story. The Lotus documents show that 1-2-3's market dominance was wholly attributable to the fact that, at the time of its market release, 1-2-3 was superior to VisiCalc, the only real competition at the time. JA 833. By 1988, "the vast majority of PC users had chosen 1-2-3 as their spreadsheet. That decision was made when 1-2-3 first came

out. . . ." *Id.* The documents explain that 1-2-3 *continued to be* successful because of the users' "investment" in "learning" and macros enabled 1-2-3 to lock in those users who first selected it over VisiCalc. *See* JA 877. As a result of the users' investment, 1-2-3 became, in the words of Lotus' own documents, "entrenched."[25] S.J. Ex. 14, 38 (Dkt. 142) [Dkt. 164, JA 78-79].

Hence, unless a new entrant in the spreadsheet market could compete for the business of the "vast majority of PC users" who initially chose 1-2-3 over VisiCalc, competition would be limited solely to new spreadsheet users, a minor portion of the market. There would have been little, if any, business to compete for; "no one" in Lotus' internal study, for example, "was using Quattro [Pro] as their first-time spreadsheet." JA 868. "Macro compatibility" according to Lotus documents, was "the most important item" to these spreadsheet users. "Macro compatibility is KEY." JA 834, 836. S.J. Ex. 14, 27 (Dkt. 142) [Dkt. 164, JA 78-79].

The Lotus documents also demonstrate that 1-2-3 users did not continue to support Lotus' product because they viewed it as superior. On the contrary, by the time of Lotus' 1990 study, "most of the 1-2-3 users did not necessarily think 1-2-3 was better, they just thought it was the product they were used to." JA 871. But because of their sunk investments, users were locked into the 1-2-3 menu command hierarchy.

Even Lotus itself was "locked in." Lotus' president Jim Manzi testified to the importance of macro compatibility in the context of a problem that arose when a version of 1-2-3, Release 2.0, was only "99-44/100 percent compatible" with previous versions. JA 715 (Manzi Depo.). Although Release 2.0 was only "off by 56/100 of a percent in terms of compatibility," Lotus was required to re-engineer "on a breakneck pace Release 2.01 to achieve 100 percent

---

[25] JA 880 (1987 Lotus marketing report stating that "there have been a number of spreadsheet products on the market that have had greater functionality than 1-2-3," but these products "have not made a significant dent in our market share due to a number of reasons, including the entrenchment of 1-2-3 [and] the investment by customer organizations in training and applications development. . . ."); JA 871 (1990 Lotus study stating "[t]here is still an entrenched 1-2-3 user base out there," noted that users were "too familiar with 1-2-3 to try to change to something else"). S.J. Ex. 27, 39 (Dkt. 142) [Dkt. 164, JA 78-79].

compatibility, because we made an awful lot of customers angry." *Id.*

The benefits of these market effects to Lotus are well-documented in the record. At the beginning of 1988, less than two months after the original Quattro product was released (and about two years before Quattro Pro was released), Lotus announced that 1-2-3 held 70% of the spreadsheet market, and that "[i]f at the end of 1988 we still own that 70%, we're home free." JA 928 (1988 Sales Speech) S.J. Ex. 24 (Dkt. 142) at 13 [Dkt. 164, JA 78-79]. In 1990, shortly after Quattro Pro was released (and after this lawsuit was filed), Lotus' President and CEO announced that 1-2-3 still held a 70% market share. JA 922 (Manzi 1990 PC Users Group Speech) S.J. Ex. 37 (Dkt. 142) at 7 [Dkt. 164, JA 78-79]. Finally, in August of 1992, at the time that Borland removed the 123-compatible mode from its products in response to the *Borland II* decision, Lotus claimed, based upon data published by the Software Publishers Association, that 1-2-3 *still* had 70% of the market. *Wall St. J.*, Aug. 13, 1992, at A5, JA 920.[26]

Lotus' brief invites the Court to ignore this record and rely instead on Lotus' extra-record assertion that the Lotus product by 1993 fell "far behind" that of Microsoft Corp., Lotus Br. at 16, as a basis for concluding that the First Circuit had a "deeply flawed understanding," *id.* Extra-record assertions are not necessary to explain Microsoft's success; the district court record contains ample explanation. Lotus declined to sue Microsoft, and hence, Microsoft was able to offer a 1-2-3 compatible spreadsheet while Borland was enjoined from doing so. *See, e.g., Borland IV* at 230, Pet.App. 39a (district court declines to rule on Microsoft's copying of the 1-2-3 menu hierarchy). A screen shot from the record below

---

[26] [Dkt. 320, 190:25]. The significance of these market effects was argued by Borland to the district court, and the district court twice held, at Lotus' urging, that lock-in with respect to macro compatibility was irrelevant to the issue of copyrightability. *Borland IV* at 233, Pet.App. 45a-46a; *Borland II* at 214, Pet.App. 124a. When a number of the user groups sought to argue before the district court, the court refused to hear them. *See* Tr. of Aug. 19, 1993 Conf. at 14 (Dkt. 406).

of the Microsoft product displaying the Lotus menus is found at A-5 in the Appendix.[27] JA 983.

D. Procedural History of the Case

The procedural history of the case is summarized in *Lotus v. Borland* at 810-12, Pet.App. 5a-10a. This lawsuit was filed on July 2, 1990. In the *Borland I* decision the district court ruled that as of March 20, 1992, almost two full years into the case, Lotus still had "not formulated for the court or for Borland its precise contentions . . . as to which elements of 1-2-3, separately or in combination, were copyrightable or were copied." *Borland I* at 98, Pet.App. 180a. Nevertheless, the district court relegated § 102(b), the list of what is not copyrightable, to mere "abstraction[s]," and held that copyright protection attaches to any and all words through which a system, process or method of operation is stated. *Id.* at 91, Pet.App. 167a.

On July 31, 1992, the district court issued its *Borland II* decision. The district court's principal holding, clearly set out in the Procedural Order that accompanied the opinion, was that "[t]he menu commands and menu hierarchy of Lotus 1-2-3 have expressive aspects and are copyrightable." Proc. Ord. (Dkt. 195) at 19. On three separate occasions after the *Borland II* decision, Borland moved to certify for interlocutory appeal the menu command hierarchy's copyrightability, but the district court denied all three motions,[28] and instead insisted on holding trials that even Lotus' counsel argued were unnecessary. *See, e.g.*, Tr. of Jan. 14, 1993 Hearing (Dkt. No. 310) at 53-54.

The first of these trials resulted in the *Borland III* decision in which the district court held that there were functional alternatives

---

[27] A more complete understanding of the reasons for Microsoft's success would require additional extra-record facts. Microsoft was able to supplant 1-2-3 only by publishing a new operating system (Windows), migrating users to that new operating system, and releasing a new spreadsheet for the new operating system before supplying Lotus with the technical information necessary to publish 1-2-3 for the new operating system.

[28] *See* Tr. of Sept. 23, 1992 Hearing (Dkt. No. 241) at 54; Tr. of Oct. 16, 1992 Hearing (Dkt. No. 242) at 15-17; Tr. of Aug. 19, 1993 Hearing (Dkt. No. 406) at 42-43.

to the 1-2-3 menu command hierarchy at the time of its creation (something Borland has never contested). In *Borland IV*, the district court held infringing Borland's "Key Reader" feature, which permitted limited macro compatibility. The district court reached this conclusion by holding that the menu command hierarchy was nothing more than "the selection and arrangement of executable operations," but was entitled to copyright protection as a "non-literal aspect of the computer program code." *Id.* at 232-33; Pet.App. 43a-45a.[29]

Finding that the district court had misconstrued § 102(b) of the Copyright Act as well as the applicable case law, the First Circuit Court of Appeals reversed the district court decisions. The First Circuit's opinion relied on the express language of the statute that forbids copyright protection for "methods of operation" and "systems."[30] Judge Boudin filed a separate opinion in which he concurred in the majority's reasoning as well as its conclusion. *Lotus v. Borland* at 821, Pet.App. at 27a-28a. Judge Boudin went on to explain that extending copyright protection to the command words at issue, as the district court had done, is at variance with the intent of Congress and is both inefficient and anticompetitive from an economic perspective.

---

[29] The district court also held that there are two methods to execute some 1-2-3 macros, "one time macro translation," and "on-the-fly" interpretation. *Borland IV* at 230, Pet.App. 38a-39a. The district court held that it is impossible to provide "on-the-fly" interpretation (the method used by Borland) without including a "copy" of the Lotus menu structure, *id.*, and held that method infringing *because* it employs the 1-2-3 menu structure, *id.* at 235, Pet.App. 48a. The district court also held that it is impossible to provide one-time translation without "copying" the 1-2-3 menu structure, but declined to decide whether one-time translation was a copyright infringement. *Id.*

[30] Lotus correctly notes that the First Circuit's opinion did not reach the merits of Borland's appeal on any of Borland's affirmative defenses, such as fair use, waiver, laches, and estoppel. Lotus Br. at 4 n. 6. If the First Circuit's decision is not affirmed, that court would have to decide those issues on remand, including Lotus' incorrect assertion that certain defenses were "abandoned." *Id.* The District Court's rejection of Borland's fair use defense is particularly erroneous in view of the intervening opinion in *Campbell v. Acuff-Rose Music, Inc.*, 114 S.Ct. 1164 (1994).

INTRODUCTION AND SUMMARY OF ARGUMENT

At the heart of this case is the question of the respective roles Congress intended for patent and copyright to play in providing protection for works such as the Lotus menu command hierarchy. In Lotus' brief, there is little acknowledgment that the dividing line between copyright and patent even is at issue. Indeed, Lotus has excised all reference to the patent law in its quotation of Art. I, § 8, cl. 8, the constitutional authorization for both sets of statutes. *See* Lotus Br. at 1, 19, 20. Similarly, Lotus has omitted any reference to the line that this Court drew between patent and copyright in its landmark decision in *Baker v. Selden*, 101 U.S. 99 (1879), even though that was the principal holding of the case.

Lotus asks this Court to hold that a software developer, through the mere assertion of copyright, can secure all the benefits of patent protection without meeting any of the statutory requirements, including "novelty," "nonobviousness," examination, and disclosure of "best mode." *See* 35 U.S.C. §§ 102, 103, 112, 131 (1995). Its principal argument in support of extending copyright protection to what the District Court repeatedly called Lotus' menu command "system,"[31] is that Congress decided that such protection was appropriate when it endorsed copyright protection for "computer programs" in 1976. Lotus' reply to the distorting effects of its claim on both patent and copyright law seems to be: Congress has spoken. *See, e.g.*, Lotus Br. at 45-49.

Lotus' argument is built upon a number of infirm foundations. The first, and perhaps most fundamental, is that the menu command hierarchy does not in fact come within the copyright statute's definition of "computer program," either on its own or as a product of the 1-2-3 code. Whatever the scope of copyright protection that Congress intended for "computer programs," this case falls outside of it. The menu command hierarchy is subject to the same regime as other works – the regime of *Baker v. Selden* and its progeny.

There are other, equally grievous flaws in Lotus' theory. Perhaps the most striking is Lotus' refusal to recognize any

---

[31] *See, e.g., Paperback* at 66, Pet.App. 232a (multiple references to the menu command "system"); *Borland II* at 213, Pet.App. 123a (the menu "system" is "fundamental" to its "functionality" as a macro language).

delimiting concept on copyright other than the "idea/expression" dichotomy. The omission is particularly notable inasmuch as it requires Lotus to read all of the terms that follow "ideas" in Section 102(b)'s limitation on copyright – including words such as "process," "method of operation," and "system," each with obvious roots in the patent law – as if they had no independent meaning whatsoever. Similarly, it leads Lotus to argue that the "goal of copyright" is to promote the "*useful* [a]rts," Lotus Br. at 24 (emphasis in original), thus seeming to overlook the fact that since 1793 the protection of the useful arts has been expressly the subject of the *patent* statute, not copyright.[32]

Also missing from Lotus' argument is any acknowledgment that the copyright and patent laws both are subject to the limitation, Constitutional in origin, that they must "ultimately serve the public good." *Fogerty,* 114 S.Ct. at 1029. Federal intellectual property rights are not free: they deprive future creators and the public at large of the free enjoyment and use of "writings" and "discoveries." The scales accordingly are weighted against the grant of a right of private monopoly: free exploitation by the public is the rule, not the exception. *Feist Publications, Inc. v. Rural Telephone Svc. Co.*, 499 U.S. 340, 349-350 (1991); *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 156-157 (1989). The creation of a private federal monopoly is appropriate only to the extent that it is necessary to ensure the promotion of new works. *Fogerty*, 114 S.Ct. at 1029; *Bonito Boats*, 489 U.S. at 150-151.

At the heart of both copyright and patent, therefore, is a careful balance between what is left in the public domain and what may be restricted to private use. Because these two intellectual property regimes cover different subjects, however, they strike the balance in different places. As Judge Boudin noted in his concurring opinion in the Court of Appeals, copyright is not unduly concerned with overprotection of writings, because the cost is relatively slight: a "mistake" would simply mean that "subsequent authors treating the same themes must take a few more steps away

---

[32] *See* Act of Feb. 21, 1793, § 1, 1 Stat. 319 (statutory subject matter of patent is "any new and useful art, machine, manufacture, or composition of matter, or any new or useful improvement [thereof]").

from the original expression." *Lotus v. Borland* at 819, Pet.App. at 23a.

In promoting the "discoveries" of "inventors," patent law strikes a different balance. Because a mistaken grant of monopoly in the most efficient or effective machine or process is more costly to the public than a monopoly on the text of *Hamlet*, patent sets the standard for a grant of private monopoly far higher: the work must be novel and nonobvious, there must be disclosure of the best mode known for carrying out the invention, and so forth. Moreover, because technological innovation most often consists of refinements and small steps, the patent law limits protection to the inventor's enumerated claims, and substantial deviation from even one of the claim's recited elements takes the later inventor outside of the patent.

In this case, Lotus' menu command hierarchy plainly was within the scope of the subject matter of the patent statute. Indeed, others, including Lotus' parent, IBM, actually have obtained patents on similar menu displays as "processes" and "systems." Lotus nevertheless contends that it should be able to obtain patent-like protection for its menu command hierarchy without any showing that its work constitutes an advance in the useful arts.

Arguing by analogy to the Copyright's treatment of "useful articles," Lotus contends that Congress could have provided narrow limits on the scope of protection for computer programs if it had chosen to do so. Lotus Br. at 30-31. In fact, however, that is precisely what Section 102(b) does, and what it was intended to do. In enacting Section 102(b), Congress made it clear that it did *not* intend for copyright protection for computer programs to cause a major shift in the balance between patent and copyright law. Indeed, it made it clear that a program's "processes" and "methods," such as the menu command hierarchy here, specifically were excluded from the scope of copyright protection. Therefore, the menu command hierarchy would be excluded under Section 102(b) even if it otherwise qualified for protection as a part of the Lotus 1-2-3 "computer program" under Section 102(a).

ARGUMENT

This Argument proceeds in two parts. Section I begins by addressing the part of the federal intellectual property scheme left out by Lotus – the patent laws. It then addresses this Court's decision in *Baker v. Selden*, and its recognition that copyright should not be used to end-run the standards required to obtain a mechanical or process patent. Section I concludes by showing that Section 102(b) was intended to incorporate this limitation on copyright, including copyright protection for computer programs.

Section II applies the terms of Section 102(b) to Lotus' claim. First, it shows that the menu command hierarchy is not protected as a "computer program" under Section 102(a), and hence is subject to the same rules as have traditionally been applied to expressive works. Second, it shows that while the menu command hierarchy may be a part of other "works of authorship" that qualify for protection under Section 102(a) – namely, the Lotus reference manual and the 1-2-3 screen display – it is an unprotected element under Section 102(b). Finally, it shows that the same result would obtain even if the "work of authorship" were considered to be the 1-2-3 "computer program."

I.

COPYRIGHT PROTECTION MAY NOT BE USED TO AVOID THE STATUTORY REQUIREMENTS OF THE PATENT LAWS.

Because copyright's central purpose is to "encourage others to build freely upon the ideas and information conveyed by a work," the Court has held that "it is peculiarly important that the boundaries of copyright be demarcated as clearly as possible." *Fogerty*, 114 S.Ct. at 1030. In drawing these boundaries, Lotus seemingly forgets that copyright is only one-half of the intellectual property scheme that Congress has devised. Copyright's contours cannot properly be marked out without considering the remainder of the Congressional scheme.

The "historic kinship" between patent law and copyright law, *Sony Corp. of America v. Universal City Studios, Inc.*, 464 U.S. 417, 439 (1984), begins with the Constitution itself. *See* U.S.

CONST. art. I, § 8, cl. 8. The Patent and Copyright Clause was enacted against the backdrop of "this Nation's historical antipathy to monopoly," *Deepsouth Packing Co. v. Laitram Corp.*, 406 U.S. 518, 530 (1972), and the patent and copyright laws share the same Constitutional limitation, which is that the monopolies they authorize "are limited in nature and must ultimately serve the public good." *Fogerty*, 114 S.Ct. at 1029.[33]

Both statutes also have incorporated the Clause's terms in setting out the scope of their respective subject matters. On the one hand, the patent statutes, commencing with the 1793 statute authored by Thomas Jefferson, have defined patent's subject matter as including any "new and useful art,"[34] thereby drawing upon the Clause's authorization to "promote the useful Arts."[35] Copyright, on the other hand, has taken as its subject the "writings" of "authors," a requirement carried forward in Section 102(a)'s protection for "works of authorship."[36] Its goal was the promotion

---

[33] *See also Sony*, 464 U.S. at 429 (copyright); *Graham v. John Deere Co.*, 383 U.S. 1, 5-6 (1966) (patent).

[34] *See* Act of Feb. 21, 1793, § 1, 1 Stat. 319. Subsequent patent statutes employed the same broad language. *See Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1980).

[35] *See* H.R. REP. NO. 1923, 82d Cong., 2d Sess. 4 (1952) (observing that under U.S. Const. art. I, § 8, cl. 8, "Congress has the power to promote the progress of useful arts by securing for limited times to inventors the exclusive right to their discoveries. The first patent law and all patent laws up to a much later period were entitled 'Acts to promote the progress of useful arts.' ").

[36] *See* 17 U.S.C. § 102(a) (1995) ("Copyright protection subsists, in accordance with this title, in original works of authorship . . . "). In employing the phrase "works of authorship," Congress sought to make it clear that it did not intend for the present Act to go to the limits of its constitutional authority. *See* H.R. REP. NO. 1476, 94th Cong., 2d Sess. 51 (1976). ("In using the phrase 'original works of authorship,' rather than 'all the writings of an author' now in section 4 of the statute, the committee's purpose is to avoid exhausting the constitutional power of Congress to legislate in this field, and to eliminate the uncertainties arising from the latter phrase"). Accordingly, some writings do not qualify as ' works of authorship" under Section 102(a). *Id.* at 51-52.

Under the 1909 Act, some jurists, including Learned Hand, had interpreted the "writings of an author" provision as reaching the constitutional limit, *see, e.g., Capitol Records v. Mercury Records Corp.*, 221 F.2d 657, 664 (2d Cir. 1955) (L. Hand, dissenting), and Congress sought to clarify that the 1976 Act did not have

of "science," a term used at the time to signify general knowledge.[37]

It is a telling statement about the thrust of Lotus' argument that it contends that the "goal of copyright" is to promote the "*useful* [a]rts." Lotus Br. at 24 (emphasis in original). Having turned a blind eye to the patent law's existence, Lotus effectively would interpret the copyright statute so as to subsume it. The patent law does exist, however; and the balance that Congress struck in the patent scheme has important consequences in establishing the limits of copyright.

A. The Patent Laws Carefully Limit the Extent to Which the "Useful Arts" May be Subject to Private Monopoly.

Three aspects of the patent statute are of particular significance for these purposes. First, the Court has construed the Clause's command to "promote the . . . useful arts" as placing important limits on the circumstances under which Congress, consistent with its Constitutional mandate, could authorize "the embarrassment of an exclusive patent." *Graham*, 383 U.S. at 9. In particular, the Court held that the Clause mandates the high bar that the patent law requires before an innovative work may be patented. The Court held, *id.* at 5-6 (emphasis in original):

---

such a reach. *See* H.R. Rep. No. 1476 at 51. ("Since the present statutory language is substantially the same as the empowering language of the Constitution, a recurring question has been whether the statutory and constitutional provisions are coextensive. . . . The bill avoids this dilemma by using a different phrase – "original works of authorship" – in characterizing the general subject matter of statutory copyright protection.") Thus, Lotus' contention that the language of Section 102(a) was intended to reach more broadly than its counterpart under the 1909 Act, *see* Lotus Br. at 20-21, presents an incomplete picture of the relationship between the two Acts. *See also* H.R. Rep. No. 1476 at 51 ("The bill does not intend either to freeze the scope of copyrightable technology *or to allow unlimited expansion into areas completely outside the present congressional intent*") (emphasis added).

[37] *See* H.R. Rep. No. 1923 at 4 ("The purpose of [the first portion of the Clause] is to promote the progress of science by securing for limited times to authors the exclusive right to their writings, the word 'science' in this connection having the meaning of knowledge in general, which is one of its meanings today.").

> The Congress in the exercise of the patent power may not overreach the restraints imposed by the stated constitutional purpose. . . . Innovation, advancement, and things which add to the sum of useful knowledge are inherent requisites in a patent system which by constitutional command must "promote the Progress of . . . useful Arts." This is the *standard* expressed in the Constitution and it may not be ignored.

Second, the Court has held that the patent system is predicated upon the public's ability to trade freely in that which does *not* meet the patent statute's standards for patentability. In *Bonito Boats*, 489 U.S. at 156, the Court observed: "[T]he efficient operation of the federal patent system depends upon substantially free trade in publicly known, unpatented design and utilitarian conceptions." The system would be undermined by a law which provided "patent-like protection" to "the functional aspects of a product which had been placed in public commerce absent the protection of a valid patent." *Id.*

In *Bonito Boats*, the Court dealt specifically with the question whether a state law which provided protection to unpatented works undermined the careful balance struck by Congress. Its analysis would apply equally, however, to the use of copyright in a way not intended by Congress that had the result of avoiding patent's novelty and nonobviousness requirements. The Court stated, *id.* at 156-157:

> Both the novelty and the nonobviousness requirements . . . provide the baseline of free competition upon which the patent system's incentive to creative effort depends. A . . . law that substantially interferes with the enjoyment of an unpatented utilitarian or design conception which has been freely disclosed by its author to the public at large impermissibly contravenes the ultimate goal of public disclosure and use which is the centerpiece of federal patent policy. Moreover, through the creation of patent-like rights, the [statute] could essentially redirect inventive efforts away from the careful criteria of patentability developed by Congress over the last 200 years.

Finally, for those inventions that overcome the hurdles to patentability, Congress has provided patentees with an explicit and powerful right to exclude, *see* 35 U.S.C. § 154 (1995), articulated in the patent case law as the right to control "use" of the patented invention, including "use" by end users.[38] The patentee's right to control "use" also includes the use of connecting to other devices – what would be known as "interface specifications" in technology jargon.[39] Copyright accords no such rights: it does not prevent the "use" of what is copyrighted, but focuses on the author's right to "reproduce the copyrighted work in copies." 17 U.S.C. § 106(1) (1995). As Judge Boudin found, however, in the part of the record that Lotus refuses to acknowledge, it is precisely the right to control "use" of the program by end users – in this case, their right to use the program in connection with Borland's product – that Lotus seeks to obtain without having shown its entitlement to patent.

B. *Baker v. Selden* and its Progeny Have Rejected the Use of Copyright to Circumvent the Requirements of the Patent Statute.

The central features of the balance that Congress has struck regarding patentability – a high bar to obtain a monopoly, and free use of that which falls below the bar – plainly are jeopardized if patent-like protection can be obtained through copyright without meeting any requirement other than "originality." Lotus has sought to minimize the significance of this disruption by suggesting that it happens all the time. Thus, Lotus has emphasized that copyright protection traditionally has been available for "useful" works such as maps, charts, and dictionaries. Lotus Br. at 24.

It is certainly true that these works are "useful" insofar as a map helps us to get from place to place, a dictionary helps us to know the meaning of words, and a telephone book helps us to

---

[38] *See, e.g., Beedle v. Bennett*, 122 U.S. 71, 78 (1887); *Coakwell v. United States*, 372 F.2d 508, 510-11 (Ct. Cl. 1967).

[39] *See, e.g., In re Hayes Microcomputer Prods. Patent Litig.*, 982 F.2d 1527 (Fed. Cir. 1992) (patent for controlling the mode of operation of a modem, or device to allow two computers to communicate over telephone lines, valid and infringed).

make telephone calls. The "usefulness" of such works does not, however, advance Lotus' claim that copyright protects the *functional* aspects of a work. The distinction is illustrated in the present Copyright Act's definition in Section 101 of "useful article": a "useful article" is an article "having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." Maps, dictionaries, and telephone books, though "useful," are not "useful articles," because they "merely . . . convey information."

Lotus' menu command hierarchy, by contrast, has "an intrinsic utilitarian function" – the words are commands that cause operations to be performed.[40] The question of how to treat works that have such an intrinsic utilitarian function is not a new one. To the contrary, it lies at the heart of *Baker v. Selden* and its progeny.

1. The Court's Baker v. Selden Decision

In *Baker*, Selden had written a book that explained a particular system of bookkeeping, and had included a series of forms, consisting of ruled lines and headings, that illustrated the system and showed how it was to be used in practice. 101 U.S. at 100. Selden urged that "the ruled lines and headings, given to illustrate the system, are a part of the book and, as such, are secured by the copyright." *Id.* at 101. Like Lotus, Selden argued that his work was "useful," and referred the Court to copyright's traditional protection of maps and charts. *Baker*, 25 L.Ed 841, 842 (argument of appellee). Also like Lotus, Selden contended that his copyright in the book gave him the right to prevent others from "using substantially the same ruled lines and headings which he has appended to his books in illustration of it." 101 U.S. at 101.

The Court rejected Selden's claim. The Court agreed that Selden had a lawful copyright in the book he wrote explaining his

---

[40] *See, e.g., Borland II* at 206-207, Pet.App. 110a ("The keystroke sequences and macro language have functionality. Typing ("inputting," in jargon) the first character of a command word invokes the command and causes the operation associated with the command word to be performed. . . . The menu command hierarchy is a fundamental part of the functionality of keystroke sequences and the macro language."); *Borland IV* at 231, Pet. App. 41a (what Lotus seeks to protect is the "selection and arrangement of executable operations").

accounting system. *Id.* at 102. Selden's copyright, however, did not give him exclusive rights to the use of his forms. The reason lay in "the difference between the two things, letters patent and copyright." *Id.* The Court gave the example of the author of a book on perspective who failed to get a patent on the mode of drawing described in the book. The author could not prevent others from using the mode of drawing that the book described, even if the book's illustrations "are reproduced in practice in the application of the art." *Id.* at 103.

As the Court emphasized, to allow the copyright in a book to extend to the methods that it illustrates would be fundamentally at odds with the patent law. Selden might or might not have been able to obtain a patent in his system; however, "it was not patented, and [now] is open and free to the use of the public." *Id.* at 104. The Court explained, *id.* at 102, in reasoning that foreshadows *Bonito Boats*:

> The copyright of the book, if not pirated from other works, would be valid without regard to the novelty or want of novelty of its subject-matter. The novelty of the art or thing described or explained has nothing to do with the validity of the copyright. To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters patent, not of copyright. The claim to an invention or discovery of an art or manufacture must be subjected to the examination of the Patent Office before an exclusive right therein can be obtained; and it can only be secured by a patent from the government.

The decision in *Baker v. Selden* thus did not, as Lotus contends, exclude only "ideas" from the scope of copyright protection. Baker was free to use not only Selden's ideas, but his forms as well: as Professor Kaplan has observed, "the [*Baker*] privilege extends to exact copies." Benjamin Kaplan, AN UNHURRIED VIEW

OF COPYRIGHT 64 (1967).[41] Under *Baker*, there is not just one delimiting concept by which the boundaries of copyright protection are set. Rather, there are two: the exclusion of ideas, and the exclusion of matter more properly the subject of patent.

2. *Cases following Baker v. Selden*

In the century since *Baker v. Selden*, courts routinely have declined to extend copyright protection to matter that falls within the scope of the useful arts.[42] In *Taylor Instrument Cos. v. Fawley-Brost Co.*, 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944), for example, the plaintiff manufactured a paper chart that provided a graphical record of hourly temperature when used in conjunction with the plaintiff's writing machine. The defendant manufactured paper charts that copied the defendant's arced and

---

[41] Indeed, the Court's holding on the matter would seem to be unambiguous. The Court held, 101 U.S. at 107 (emphasis added): "The conclusion to which we have come is that blank account-books *are not the subject of copyright*; and that the mere copyright of Selden's book did not confer upon him the exclusive right to make and use account-books, *ruled and arranged as designated by him and described and illustrated in said book*."

Lotus nonetheless has urged that the case should be read as turning on the extent of the similarities between Baker's and Selden's forms; according to Lotus, Baker would have been found to have infringed if he had copied Selden's forms more closely. Lotus Br. at 34-36. The Court's decision, however, cannot be squared with Lotus' analysis. Indeed, Lotus has had to rely on facts outside the opinion even to make the argument, because one cannot tell how closely Baker followed Selden from the decision itself. It simply was not a focus of the Court's analysis: the Court was concerned with *Selden's* words, not with Baker's, and with Selden's ability to monopolize his system by monopolizing the forms. Lotus' analysis would render the Court's analysis of the distinction between patent and copyright superfluous, and simply cannot be reconciled with many portions of the opinion. *See, e.g., id.* at 103 (copyright "cannot give to the author an exclusive right to the methods of operation which he propounds, or to the diagrams which he employs to explain them, so as to prevent an engineer from using them whenever occasion requires").

[42] *See, e.g., Brief English Sys., Inc. v. Owen*, 48 F.2d 555, 556 (2d Cir. 1931) (copyright does not provide exclusive right to use system of shorthand; "the way to obtain the exclusive property right to an art, as distinguished from a description of the art, is by letters patent and not by copyright"), *cert. denied*, 283 U.S. 858 (1931). *See also Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182 (7th Cir. 1944) (method of business reorganization), *cert. denied*, 322 U.S. 755 (1944); *Affiliated Enters. v. Gruber*, 86 F.2d 958 (1st Cir. 1936) (promotional system).

circular lines in order to make them "compatible" with the defendant's machine. Applying *Baker v. Selden*, the court concluded that the chart "neither teaches nor explains the use of the art. It is an essential element of the machine; it is the art itself." *Id.* at 100. As such, it was properly protected, if at all, by obtaining a utility patent.

Lotus nonetheless has contended that this aspect of *Baker v. Selden* and its progeny was reversed by the Court in *Mazer v. Stein*, 347 U.S. 201 (1954). *See* Lotus Br. at 25. The *Mazer* decision, however, does not support Lotus' claim. The question presented in *Mazer* was whether there could be overlap between the copyright law and *design* patents, which reward creators for their "ornamental design[s]." 347 U.S. at 216. As *Mazer* found, overlap between design patents and copyright long has been recognized by the courts. *Id.* at 215 n.33. Indeed, in *Baker v. Selden* itself, the Court expressly stated that "[o]f course, these observations are not intended to apply to ornamental designs." 101 U.S. at 103. The Court recognized that with ornamental designs, as with copyrighted works, "their form is their essence"; allowing a monopoly in such ornamental features accordingly imposed no undue cost on the public. *Id.*

As *Mazer* expressly recognized, a *different* result has obtained, beginning with *Baker* itself, in cases dealing with utility patents. Thus, *Mazer* noted that while courts had found an area of overlap between design patents and copyright, "a different answer has been given by the courts" with respect to "the mechanical patent law and copyright laws." 347 U.S. at 215 n.33. Of the two cases cited with approval by the Court for this point, one was the *Taylor Instrument* case. Thus, far from providing support for Lotus' theory of overlap, *Mazer* reconfirmed the continuing vitality of *Baker v. Selden's* distinction between patent and copyright. Selden's forms and Taylor's charts were not outside of copyright protection because they were "ideas," but because they were within the province of the "useful arts," and the extension of protection to these works would have undermined the patent regime.

C. Section 102(b) Incorporates the Limitation on Copyright Protection For Utilitarian Functions.

According to Lotus, Section 102(b) was intended to do no more than exclude "idea[s]" from the scope of copyright protection. Invoking the principle of *noscitur a sociis*, Lotus contends that the seven words listed after "idea" in the statute are "undifferentiated," or, in plainer English, redundant. Lotus reasons that "process[es]" or "method[s] of operation" could be construed so broadly as to cover computer programs themselves; that Congress intended for computer programs to be copyrightable; and that accordingly these terms (and the remaining terms of Section 102(b)) should not be given any independent meaning whatsoever. Lotus Br. at 29-30. Lotus urges that when the Court reads "method of operation," it should substitute "idea" instead. *Id.* at 29.[43]

The drastic surgery that Lotus proposes in interpreting Section 102(b) is unwarranted. Generally, of course, the Court "will avoid a reading which renders some words altogether redundant." *Gustafson v. Alloyd Co., Inc.*, 115 S. Ct. 1061, 1069 (1995). Here, the remaining words of Section 102(b) are neither obscure nor redundant: they are drawn directly from the patent law. Moreover, the legislative history shows that they were central to the limits that Congress sought to impose on the scope of copyright protection for computer programs. Far from being awkward surplusage, therefore, they are critical to ascertaining the boundaries on copyright protection of computer programs under Section 102(b).

1. The Terms in Section 102(b) are Drawn From the Patent Law.

In invoking *noscitur a sociis*, Lotus may seem to suggest that "process" and "method of operation" are words "of obscure or

---

[43] This argument also is central to the two *amicus* briefs filed on Lotus' behalf. *See* Brief *Amicus Curiae* of American Intellectual Property Law Association at 8 n.10 (Section 102(b) terms are used "interchangeably"); *Amicus Curiae* Brief of Digital Equipment Corporation, The Gates Rubber Company, Intel Corporation and Xerox Corporation at 10 n.8 ("unless the context dictates otherwise," the amici would "refer to all of the categories of unprotectable matter listed in Section 102(b) collectively as ideas").

doubtful meaning." *Russell Motor Car Co. v. United States*, 261 U.S. 514, 520 (1923). Nothing, however, could be further from the truth. As noted previously, the patent statutes commencing in 1793 all extended protection to any "new and useful art." *Chakrabarty*, 447 U.S. at 308-09. In 1952, when the patent laws were recodified, Congress made one change to Jefferson's language: it replaced the word "art" with "process." *Id.* at 309. "Process" thus lies at the very heart of the patent statute; and, if one were to carry forward *Baker v. Selden's* exclusion from copyright of the "useful arts," the way to express that in the present-day language of the patent laws is to use the term "process."[44]

The patent-law derivation of the terms "method of operation," and "system" is no more difficult to find. Both "method of operation" and "system" were used in *Baker v. Selden* to describe that which "is the province of letters patent, not of copyright." *See* 101 U.S. at 102-104.[45] Indeed, since *Baker v. Selden*, copyright cases often have used the term "system" to describe matter that

---

[44] *See* 35 U.S.C. § 101 (1995). Although only recently codified, "process" and "useful art" have been treated as largely equivalent terms in the patent law for many decades. *See, e.g., Expanded Metal Co. v. Bradford*, 214 U.S. 366, 382 (1909) ("The inventor of a new and useful art is distinctly entitled to the benefit of the statute as well as he who invents a machine, manufacture, or composition of matter. The word 'process' has been brought into the decisions because it is supposedly an equivalent form of expression, or included in the statutory designation of a new and useful art.").

[45] The terms "process" and "method" often are used interchangeably in the patent law. *See, e.g.*, 35 U.S.C. § 100(b) (1995) ("The term 'process' means process, art or method"). Methods "of operation," however, may be considered to have a more specific meaning, as reflected in the Court's decision in *Expanded Metal*, 214 U.S. at 382. Prior to *Expanded Metal*, it had been unresolved, as counsel in that case put it, whether "processes involving mechanical operations, as distinguished from chemical reactions or elementary changes, are unpatentable." *Expanded Metal*, 53 L.Ed. 1034, 1035. After *Expanded Metal* upheld their patentability, such method claims often have been described as a method "of operating" a particular machine. *See, e.g., Application of Moreton*, 288 F.2d 708, 709 (C.C.P.A. 1961) (claim for a "method of operating the hydraulic system of an aircraft"); *Application of Horvath*, 211 F.2d 604, 606 (C.C.P.A. 1954) (claim for a "method of timing succeeding toaster operations"); *Application of Schutt*, 210 F.2d 293, 294 (C.C.P.A. 1954) (claim for a "method of operating a continuously cycling automatic concrete block machine").

falls within the ambit of patent rather than copyright. *See, e.g., Affiliated Enters. v. Gruber*, 86 F.2d 958 (1st Cir. 1936) (promotional system); *Brief English*, 48 F.2d at 556 (system of shorthand); *Amberg File & Index Co. v. Shea Smith & Co.*, 82 F. 314, 315 (7th Cir. 1897) (indexing system); *Griggs v. Perrin*, 49 F. 15 (C.C.N.D.N.Y. 1892) (shorthand system). In employing the term, Congress plainly intended to ratify and codify the line drawn in these cases. *See Davis v. Michigan Dept. of Treasury*, 489 U.S. 803, 813 (1989).

The derivation of "discovery" is, if anything, even more apparent: the term can be found in the Patent and Copyright Clause itself, in the portion of the Clause for which Lotus has substituted an ellipsis. As noted earlier, the Clause authorizes Congress to protect "inventors" in their "discoveries." Art. I, § 8, cl. 8. It would seem that Congress could not have put any more emphatically its intent to maintain the line between patent and copyright. The term "principle" also may be found to have roots in the patent law. Indeed, "principle" long has been used as the patent equivalent of "ideas," for which no patent can be obtained. *See e.g., LeRoy v. Tatham*, 55 U.S. 156, 174-175 (1852) ("A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right").

The derivation of "concept" and "procedure" is more obscure. The ordinary meaning of "concept" is an "abstract or generic idea," and the legislative history suggests that the term was intended to have that meaning.[46] *See* S. REP. NO. 983, 93d Cong., 2d Sess. 107-108 (1974) (in earlier version of proposed Section 102(b) which excluded "plans," "plans" were "distinguished from . . . the mental concept"). No patent can be obtained for a "concept" lacking a physical embodiment. *See, e.g., Voightmann v. Perkinson*, 138 F. 56, 57 (7th Cir. 1905); *Application of Hortman*, 264 F.2d 911, 913 (Cust. & Pat. App. 1959). A "procedure" in ordinary use means "a particular way of accomplishing something or of acting," or "a series of steps followed in a regular definite order,"

---

[46] WEBSTER'S NINTH NEW COLLEGIATE DICTIONARY 272 (9th ed. 1988) ("WEBSTER'S").

WEBSTER'S, *supra* at 937, and it appears to have been used in that way in patent cases, either in describing steps in a process or method or sometimes as the equivalent of "process" itself.[47]

In short, the terms in Section 102(b) are far from "undifferentiated." They establish, by their terms, Congress' clear intent that copyright should not be permitted to substitute for or interfere with the subject matter of patent – both that which is unpatentable, such as principles, and that which is, such as processes and methods of operation. Indeed, in attempting to lump these terms together as undifferentiated "ideas," Lotus would undo much of patent law, in which the distinction between "processes" (or "methods of operation") and "ideas" (or "principles") is as central as the distinction between "expression" and "ideas" is to copyright. *See, e.g., Diamond v. Diehr*, 450 U.S. 175, 182-185 (1981) (contrasting patentable "processes" with unpatentable "ideas"). "Process" and "method of operation" for these purposes are the *opposite* of "ideas" – a fact of which Congress plainly was aware.[48]

---

[47] *See, e.g., Saranac Automatic Mach. Corp. v. Wirebounds Patents Co.*, 282 U.S. 704, 708, 715 (1931); *Graver Tank & Mfg. Co., Inc. v. Linde Air Prods. Co.*, 336 U.S. 271, 278 (1949); *U.S. Indus. Chems., Inc. v. Carbide & Carbon Chems. Corp.*, 315 U.S. 668, 676-77 (1942); *Application of Drummond*, 302 F.2d 761, 764 (C.C.P.A. 1962); *Application of Orsini*, 158 F.2d 286, 287 (C.C.P.A. 1946); *Raffold Process Corp. v. Castanea Paper Co.*, 98 F.2d 355, 358-359 (3d Cir.), *cert. denied*, 305 U.S. 635 (1938).

During the time that the copyright revision legislation was pending, the Court handed down its decision in *Gottschalk v. Benson*, 409 U.S. 63 (1972). In *Gottschalk*, the Court specifically used the term "procedure" in its definition of an algorithm. *See* 409 U.S. at 65 ("A procedure for solving a given type of mathematical problem is known as an 'algorithm' "). At the time Section 102(b) was enacted in 1976, therefore, "procedure" might have been understood as applying, *inter alia*, to unpatentable algorithms. *See also id.* at 67 ("abstract intellectual concepts" are not patentable).

[48] *See Copyright Law Revision: Hearings Before the Subcomm. on Courts, Civil Liberties, and the Admin. of Justice of the House Comm. on the Judiciary*, 94th Cong., 1st Sess. 2223 (1975) (proposal by Computer & Business Equipment Manufacturers Association to excise "plan, procedure, process, system, method of operation," leaving "idea, concept, principle, or discovery" as uncopyrightable subject matter); *id.* at 334 (proposal by Information Industry Association to amend 102(b) to allow protection for a "collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes").

Nor do the terms in Section 102(b) denote mere "abstractions," as the district court suggested. *Borland I* at 91, Pet. App. 167a. The line between what is copyrightable and what is not has never been drawn at the point at which words are first attached to abstract thought. *But see id.* "Expression" in the copyright law has regularly been construed to include "abstractions" such as the detailed aspects of the plot of a play,[49] while at the same time the textual labels on Selden's forms were held not to be protected. Similarly, some of the § 102(b) words from the patent law denote what the district court referred to as an "abstraction" (*e.g.*, "principle"); the disclosure of the "best mode" for a patent claim covering Lotus' method of operation, however, would doubtless cover the words themselves. 35 U.S.C. § 112 (1995).

2. *A "Computer Program" is not a "Process" or "Method of Operation."*

Lotus seems to argue that, even if the terms of Section 102(b) have a plain meaning under the patent laws, they should not be given that meaning because to do so would be inconsistent with Congress' intent to extend protection to computer programs. Lotus Br. at 29-30. The legislative history, however, shows that Congress saw no such conflict. Indeed, it is evident that the exclusion of "processes" and "methods of operation" in Section 102(b) was meant specifically to *limit* the protection afforded to computer programs. Thus, the House Report states that Section 102(b) was intended, among other things, to make clear that the "processes or methods embodied in the [computer] program are *not* within the scope of copyright law." H.R. REP. 1476 at 57 (emphasis added).

---

[49] *See, e.g., Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931); *Kouf v. Walt Disney Pictures & Television*, 16 F.3d 1042, 1045 (9th Cir. 1994) (comparing similarities of ideas and expressions in two works, with "plot" as an expressive element).

The House Report indicates that Section 102(b) was enacted in part in response to "some concern [that] has been expressed" regarding the protection of program processes and methods. *Id.* This concern was voiced principally during hearings before the Senate Subcommittee in 1967 on a predecessor revision bill, S.597. During these hearings, the report of the Interuniversity Communications Council (more commonly referred to as EDUCOM), particularly cautioned that extending copyright protection to a program's "process" would "amount to giving programs a breadth of protection similar to that accorded by patent, but without the safeguards and limitations that rightly surround the grant of a patent." *Copyright Law Revision: Hearings Before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on The Judiciary*, 90th Cong., 1st Sess. 572 (1967). EDUCOM concluded that copyright protection could be applied to computer programs, but only if narrowly cabined, *id.* (emphasis added):[50]

> If the process embodied in a computer program ought not to be aggrandized through copyright, it might still seem plausible to allow a narrower copyright – one that would confer upon the copyright proprietor the exclusive right to replicate *the instructions themselves* . . . But it becomes evident that this right must be carefully circumscribed.

---

[50] It appears that the EDUCOM report and its ensuing recommendations may have been a source of what ultimately became Section 102(b). During EDUCOM's oral presentation, the chairman of the Senate subcommittee asked EDUCOM to propose language that would solve its concerns. *Id.* at 562, 565. EDUCOM reported back with two proposals, one of which would have amended Section 106 relating to the rights of copyright owners as follows: "*Provided, however,* that nothing in this title shall be construed to give the owner of copyright the exclusive right to any idea, process, plan, or scheme embodied or described in the copyrighted work . . . " *Id.* at 1059.

S. 597 ultimately was not reported out of subcommittee. The next session, however, a nearly identical bill, S. 543 (91st Cong.), was reported to the Judiciary Committee with the addition of a proposed new Section 102(b). This amendment, which appears to be based in part on the EDUCOM proposal, is identical to the enacted version of Section 102(b) except for the inclusion in the bill of the word "plan." *See* 1 KAMINSTEIN LEGISLATIVE HISTORY PROJECT 41-42 (1981), *citing* S. REP. NO. 1219, 91st Cong., 2d Sess. (1970) (Committee Print).

The same distinction was drawn in the *Final Report of the National Commission on New Technological Uses of Copyrighted Works* (1978) ("CONTU Report"), which recommended the changes enacted in the 1980 Software Amendments. The Report assured Congress that the line between programs and processes remained intact. Accordingly, "one is always free to make a machine perform any conceivable process (in the absence of a patent)." *Id.* at 20. The only program elements identified as protected by copyright were the programmer's writings: the source code, the object code into which it was translated, and possibly the programmer's flow charts (perhaps as a pictorial or graphic work). *Id.* at 21 & n.109.

For more than a century, *Baker v. Selden* and its progeny have held that copyright protection should not be allowed to intrude upon and disrupt the careful balance that has been struck under the patent laws concerning when to allow a private monopoly in the "useful arts." Legislating against that background, Congress codified that line of decisions in Section 102(b), and exhibited particular concern that copyright protection for computer programs not be used to undermine the patent regime. In excluding "ideas" from copyright protection under Section 102(b), Congress sought to ensure that authors were not given too broad a monopoly relative to other authors;[51] but in excluding "processes" and "methods" from Section 102(b), Congress also has sought to ensure that authors, by using copyright to avoid the restrictions of the patent law, cannot obtain an unwarranted monopoly relative to *inventors*. As *Bonito Boats* teaches, to allow an inventor to circumvent these restrictions, and obtain a monopoly on an unpatented utilitarian work through the simple expedient of calling itself an "author," is to jeopardize the integrity of the patent system as a whole.

---

[51] *Feist*, 499 U.S. at 349-50 (the idea/expression distinction is intended to ensure that an author is not overcompensated relative to later authors, by "assur[ing] authors the right to their original expression, but encourag[ing] others to build freely upon the ideas and information conveyed by a work"). *See generally* William M. Landes and Richard A. Posner, *An Economic Analysis of Copyright Law*, 18 J. OF LEGAL STUD. 325, 333 (1989) (the copyright law's dichotomy between idea and expression can be understood as an attempt to promote economic efficiency by preventing overcompensation).

II.

LOTUS' MENU COMMAND HIERARCHY IS UNPROTECTED UNDER SECTION 102(b) OF THE COPYRIGHT ACT.

In its report to Congress, CONTU noted that the distinction between the protected elements of a computer program and those excluded as "processes" or "methods of operation" would not always "shimmer with clarity." CONTU Report at 18. Nonetheless, CONTU advised Congress that "[t]o attempt to establish such a line in this report written in 1978 would be futile." *Id.* at 22. CONTU accordingly advised Congress that "[s]hould a line need to be drawn to exclude certain manifestations of programs from copyright, that line should be drawn on a case-by-case basis by the institution designed to make fine distinctions – the federal judiciary." *Id.* at 22-23.

Ironically, Lotus now chastises the Court of Appeals for making precisely such a determination, on the ground that the line already has been drawn by Congress. Lotus Br. at 45-49. In fact, however, the Court of Appeals was acting in accordance not only with what Congress expected, but what the Constitution and this Court have required. This Court repeatedly has stressed that the purpose of the copyright laws is not to "maximiz[e] the number of meritorious suits for copyright infringement," but rather to "enrich[] the general public through access to creative works." *Fogerty*, 114 S. Ct. at 1029. To justify what the Court has referred to in the patent context as the "embarrassment" of a private monopoly, *Bonito Boats*, 489 U.S. at 148, the copyright law must encourage and reward authors, but only insofar as it "serve[s] the cause of promoting broad public availability of literature, music, and the other arts." *Fogerty*, 114 S. Ct. at 1029, quoting *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975). Where the literal terms of the statute are ambiguous, the Copyright Act must be construed in light of this basic purpose.[52]

---

[52] *Twentieth Century Music Corp.*, 422 U.S. at 156; *Sony*, 464 U.S. at 432. *Cf. Graham*, 383 U.S. at 5 ("The Congress in the exercise of the patent power may not overreach the restraints imposed by the stated constitutional purpose").

A. The Lotus Menu Command Hierarchy is Not a "Computer Program" Under Section 101.

Much of Lotus' brief rests on the proposition that the menu command hierarchy is a "computer program" (or a portion of one) within the meaning of Section 101. Section 101 defines a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1995). Lotus argues that not just the 1-2-3 program's code, but the menu command hierarchy itself, "fits" this definition, in that it could be described as a "set of statements or instructions" that can be used in a computer to accomplish "a certain result." Lotus Br. at 30. Alternatively, Lotus suggests that the command hierarchy is part of the "computer program" because it "is generated by the statements contained in the program's 'source code.' " *Id.* at 7.

Lotus' initial argument – that the menu command hierarchy generated by the 1-2-3 program is itself another "computer program" – is not consistent with the ordinary way in which the term "computer program" is used. Indeed, Lotus does not claim that it is. Users looking at the command words and other symbols arrayed on a computer screen would not describe these figures as a "computer program," but rather as the means for *accessing* a computer program. Similarly, the CONTU Report that recommended the definition of "computer program" adopted in Section 101 never remotely suggested that the definition that it proposed would extend to portions of a screen display. Rather, the CONTU Report uses "computer program" in the way that it is customarily used: to describe source code and object code.[53]

---

[53] *See* CONTU Report at 21 n.109 (identifying source code and object code as "programs"); *see also* CONTU Transcript of Sept. 15, 1977 CONTU meeting at 78 (explaining that phrase "directly or indirectly" is used in the definition of "computer program" because of intent to cover both object code and source code). The CONTU Report is ambiguous as to whether "flow charts," which are used by programmers to diagram the program's sequence of steps before actually writing the code, also might be protected as graphical works. *See* CONTU Report at 21 & n.109.

It is true that the Lotus menu commands may be described as a program *language*,[54] but that only serves to highlight the way in which it differs from a computer *program*. Lotus' own brief emphasizes that it is only after a user combines "an entire sequence of keystrokes" that the computer will "actually perform an operation" – in other words, will "bring about a certain result." *See* Lotus Br. at 8-9. Lotus describes the entry of each such "sequence of keystrokes" as "an instruction." *Id.* A set of such keystroke sequences, in turn, would seem to fit the language of Section 101, which requires "a *set* of statements or instructions." That is how Lotus' users' manual defines a "macro": it is a "set of instructions" made up of "a sequence of keystrokes and commands." 1-2-3 Ref. at 166.

A "macro" thus may be a "set of statements or instructions," *i.e.*, a "computer program," but the menu command hierarchy is not. Until the user enters the keystrokes in a combination that will cause the computer to perform an operation, there has not been the entry of a "set" of instructions that will cause the computer to "bring about a certain result." Indeed, as Lotus itself insists, until the user enters a keystroke sequence into the computer, there has not been even a single "instruction." Lotus Br. at 8. The menu commands thus may provide the building blocks from which a "computer program" can be built; but the creative effort that transforms these building blocks into a computer program is provided by the user, not by Lotus.[55]

Lotus' alternative argument for coming within the definition of "computer program" under Section 101 is that the menu command hierarchy is a part of the program because it is "generated"

---

[54] *See, e.g., Borland II* at 206, Pet.App. 109a (describing the use of keystroke sequences as a "macro language").

[55] The importance of the requirement of a set or series of instructions is borne out in the ordinary dictionary definition of a computer program. *See, e.g.*, WEBSTER'S, *supra* at 940 ("program" defined as "a sequence of coded instructions that can be inserted into a mechanism (as a computer)"). More specialized dictionaries also are to the same effect. *See, e.g.*, IBM DICTIONARY OF COMPUTING, 131 (10th ed. 1993) (a "computer program" is "[a] sequence of instructions suitable for processing by a computer"); MICROSOFT PRESS COMPUTER DICTIONARY 90 (2d ed. 1994) (a "computer program" is "[a] set of instructions in some computer language, intended to be executed on a computer to perform a useful task").

by the program. Lotus Br. at 7. This is not an argument on which Lotus has rested much weight, and with good reason. As Lotus' own brief attests, the 1-2-3 "statements" to the computer are "contained in the program's 'source code,' " and the menu command hierarchy is "generated" by those statements. *Id.*[56] In the language of Section 101, the source code is the "set of statements or instructions" that constitutes the "computer program"; the menu command hierarchy (as well as the rest of the screen display) that it generates is the "certain result."[57]

The question of copyright protection for the screen display generated by the 1-2-3 program must be assessed independently from Lotus' copyright in the program itself. Lotus' argument regarding the intended breadth of protection for computer programs, *see, e.g.*, Lotus Br. at 30-31, accordingly is irrelevant. It is as part of the 1-2-3 screen display, or as part of the Lotus users' manual, that the menu command hierarchy must be a protected element; it is not part of the 1-2-3 "computer program" within the meaning of Section 101 and 102(a).

B. The Menu Command Hierarchy is an Unprotected Element of the Lotus Users' Manual and the Lotus Screen Display.

Lotus has contended, somewhat cynically, that "[w]ere these words fixed upon a series of printed pages, instead of appearing on

---

[56] *See also Borland II* at 209, Pet.App. 114a ("The menu command hierarchy is part of the 1-2-3 program's output.").

[57] *See also* 1 Paul Goldstein, COPYRIGHT § 2.15.3.1, at 2:200, 2:202 (2d ed. 1996) (The suggestion that copyright in the computer program would extend to the screen display would "dramatically depart" from the statutory definition of computer programs. Accordingly, "[c]opyright in a computer program that produces certain screen displays will protect the program but will not prohibit a competitor from independently designing a program that produces the same screen displays."). If the language of the statute were in any way ambiguous, this interpretation would be confirmed by the CONTU Final Report. The report made it clear that copyright protection for the computer program was to be determined entirely apart from the question of copyright protection for the output or "certain results" generated by the program. *See* CONTU Report at 21. The results generated by the program thus have to qualify separately for copyright protection; their status does not affect the program's copyright.

a computer screen, there could hardly be a question that they would be protected by copyright." Lotus Br. at 17. In fact, however, the words *are* fixed upon a series of printed pages – the pages of the Lotus users' manual (and dozens of other books about 1-2-3 as well). Whether as part of Lotus' users' manual, however, or as part of the Lotus screen display, the Lotus menu command hierarchy is unprotected; it is a system or method of operation under Section 102(b).

1. The Lotus Reference Manual

Just as Selden prepared a book for users of his copyright system, Lotus has prepared a book for users of the 1-2-3 program. In the 1985 version of Lotus' manual, 138 pages are devoted to explaining the functions performed by different menu commands; another 60 explain how to use the commands to build macros. 1-2-3 Ref. at 28-217 (Dkt. 158) [Dkt. 164, JA 78-79]. At the beginning of the chapter committed to menu commands, there is a "functional summary" that lists approximately 150 commands, with a one-line description of the tasks that each command performs. *Id.* at 28-37. The summary illustrates that the Lotus menu commands are not a "structured dialogue" with the user. Rather, the command combinations simply are listed ("/Range Format Text") with a description of the operation they perform ("display formulas instead of values"). *Id.* at 29. The summary is not even organized by hierarchy commands, but by user interest.

Nothing could be clearer from Lotus' users' manual than that the menu commands do not illustrate or explain the method of operation – they *are* the method of operation.[58] All of the terms used to explain the menu commands are unambiguously functional: the Range commands "manipulate" ranges of cells, *id.* at 62; the File commands "save" worksheets, *id.* at 86, and so forth. Lotus cites the District Court's conclusion that there were "expressive" elements to the commands and their arrangement, Lotus Br. at 41, but it does not explain, much less defend, the District Court's

[58] Lotus' contention that Borland did not argue below that the command hierarchy is a method of operation, Lotus Br. at 43 n.65, is belied by the record. Borland 1st Cir. Br. at 39, 50; Borland S.J. Br. at 29-113.

rationale in reaching this conclusion. The District Court found the command hierarchy "expressive" because a "satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure." *Borland II* at 217, Pet.App. 10a. The fact that there is more than one way to achieve a particular function, however, is pertinent to the question whether the system can be *patented*, not whether it is the proper subject of copyright. As any overseas traveler can attest, there are many seemingly satisfactory ways of arranging the configuration of electrical sockets; yet that does not render the shape of any particular socket an "expression" protectable under copyright.

The Court's decision in *Baker v. Selden* is instructive on this point. The Court did not find that there were only a limited number of ways to practice double-entry bookkeeping; rather, it held that Baker had a right to practice Selden's way. Similarly, although the Court found that Selden's forms were expressive in the sense that they "illustrated his method," 101 U.S. at 104, that did not suffice to render the forms a protected element of Selden's copyright.

2. The Lotus 1-2-3 Screen Display

As the District Court's opinions below reflect, it is difficult to describe the command hierarchy without referring to its functionality, or even without using the specific terms excluded under Section 102(b). *See, e.g., Borland IV* at 231, Pet.App. 41a (menu tree may be viewed as the "selection and arrangement of the executable operations in Lotus 1-2-3"); *Paperback* at 65, Pet.App. 229a ("the exact hierarchy – or structure, sequence and organization – of the menu *system*") (emphasis added). Lotus labors under the same difficulty. Thus, its users' manual describes 1-2-3's commands as "organized into a multi-level menu system." JA 528 (Raburn Decl.) and JA 533-34 (Ex. A) [Dkt. 164, JA 78-79].

Similarly, in Lotus' European patent application for a user interface, it describes a "control hierarchy" as "divid[ing] controls, assign[ing] each control a label, display[ing] the labels according to a design hierarchy, and operat[ing] on the controls according to a functional hierarchy." JA 838. In the language of patent law, the recited claim is for "a *method* of providing a plurality of controls." JA 841 (emphasis added). The "method" is comprised of four

steps, the last of which is "*operating* on said controls according to said functional hierarchical groups." *Id.* (emphasis added). A command hierarchy is, in short, "a method of operati[on]."

The ease with which the menu command hierarchy fits patent terms such as "system" and "method of operation" is not coincidental. Indeed, this case poses squarely the question whether copyright extends to the subject matter of utility patents, because the Patent Office has issued patents on menu command hierarchies like Lotus'. Thus, for example, IBM, Lotus' parent, obtained a patent in 1983 for a "display menu" on a word processing program. JA 856. Similarly, a spreadsheet software maker was able to obtain protection for a system employing a command hierarchy similar to Lotus'. JA 842. S.J. Ex. 15, 16, 23 [Dkt. 164, JA 78-79].

In arguing for overlapping protection, Lotus has identified no ready means by which its proposed interference with patent could be cabined. Indeed, Lotus' effort to distinguish the menu command hierarchy from the Court of Appeals' analogy to the VCR provides a glimpse of just how vast is the subject matter potentially affected by Lotus' rule. Lotus asserts that the most "fundamental" difference between its array of menu commands and the "user interface" of a VCR is that a VCR interface consists of roughly six buttons, whereas the number of commands using the 1-2-3 hierarchy exceeds four hundred. Lotus Br. at 17 n.27. To begin with, Lotus' distinction is an artificial one. Many users do not operate a VCR just by pushing the VCR's buttons; they also use a remote control that operates through the selection of program-generated menus on the television screen (*e.g.*, to delay-record a television program). Sony VCR Operating Instructions, Ex. 5 to Borland S.J. Br. (Dkt. 142) [Dkt. 164, JA 78-79]. Nothing that Lotus has proposed would distinguish its menu command hierarchy from these VCR controls.

More fundamentally, Lotus provides no meaningful conceptual basis why the arrangement of six buttons is not copyrightable, but four hundred buttons is. By Lotus' analysis, the arrangement of buttons on the control panel of a commercial jet certainly would be the subject of copyright. What about the sixty to eighty buttons on a typical computer keyboard? The twenty to thirty buttons on the typical business telephone? Menu command hierarchies generated by software are everywhere: on gasoline pumps, automatic teller machines, photocopy machines. Lotus proposes that manufacturers

should be able to obtain a monopoly under copyright on the arrangement of these controls, and force users to learn a new pattern of controls with every different product that they use. From the point of view of "the public good," *Fogerty*, 114 S. Ct. at 1029, the Court of Appeals aptly summarized such a result with one word: absurd. *Lotus v. Borland* at 818, Pet.App. 20a.[59] It is no wonder that Lotus' sole amici are four hardware manufacturers who were early entrants (often with dominant market share) in their markets. *See* Brief *Amicus Curiae* of Software Forum In Support of Respondent at 13-15.

Lotus attempts to analyze away the command hierarchy's essential functionality by describing it as a freestanding "literary work" unconnected to the user interface of which it is a part. *See* Lotus Br. at 22 (the command hierarchy is a "literary work" under Section 102(a)). *Compare id.* at 7 (the command hierarchy "is part of what is typically called the 'user interface' "). Lotus provides no justification for treating one portion of the Lotus screen display as a discrete "work of authorship," any more than the Court treated the headings and arrangement of words on Selden's forms as a separate work of authorship. In fact, Lotus' approach is highly distorting. By simply ignoring both the remainder of the screen display (such as the long prompts and on-line help texts), Lotus assumes away the awkward fact that the hierarchy is treated as the matter to be explained, and not the explanation; and by stripping

---

[59] Indeed, Lotus' proposal would directly undermine Congress' intent in establishing examination requirements before a patent can be issued. Prior to the 1836 amendment to the patent laws, the granting of patents – like copyrights today – was a purely ministerial function. With rapid industrialization, however, Congress found that the system had become unworkable. *See* S. REP. NO. 338, 24th Cong., 1st Sess. at 3 (1836) ("The country becomes flooded with patent monopolies, embarrassing to bona fide patentees . . . and not less embarrassing to the community generally, in the use of even the most common machinery and long-known improvements in the arts and common manufactures of the country. . . . Out of this interference and collision of patents and privileges, a great number of lawsuits arise, which are daily increasing in an alarming degree, onerous to the courts, ruinous to the parties, and injurious to society."). The "interference and collision of privileges" likely to result from Lotus' claim would pose the same threat to efficient economic growth that Congress sought to prevent with the reforms to the patent system in 1836.

away the menu's words from the "virtual buttons" to which they are attached, Lotus pretends that the case is not as much about machines as any other case about "VCR buttons, automobile gearshifts, or any other type of machine part." Lotus Br. at 17. No amount of wishing, however, will make the command hierarchy into something that it is not. The menu command hierarchy is unprotected as a "system" or "method of operation" under Section 102(b), whether as part of the Lotus users' manual or the Lotus screen display; and it does not become transformed into something else by considering it a separate literary work under Section 102(a).

C. Even if Part of the Computer Program for Purposes of Section 102(a), the Menu Command Hierarchy is an Unprotected Method of Operation under Section 102(b).

The command hierarchy would be excluded matter under Section 102(b) even if the work of authorship under Section 102(a) were the Lotus "computer program" itself.[60] Lotus does not seriously dispute that the menu command hierarchy can be described as the "method of operating" the 1-2-3 program. Indeed, it is difficult to imagine what the method of operating 1-2-3 would be if it is *not* through use of the command hierarchy. In the video that Lotus prepared in connection with the summary judgment phase of proceedings in the District Court, its spokesman referred repeatedly to "using the menus" to "perform operations." *See* Dkt. No. 127 [Dkt. 164, JA 78-79] (Borland Response Video containing

---

[60] Lotus has argued that computer programs "enjoy copyright protection under the same principles governing other forms of original expression in literary works." Lotus Br. at 31. *Compare* Kenneth A. Liebman, et al., *Back to Basics: A Critique of the Emerging Judicial Analysis of the Outer Limits of Computer Program "Expression,"* 2 COMPUTER LAW., Dec. 1985, at 1, 8 (quoting Prof. Miller) ("The end purpose of a computer program is to achieve a utilitarian result, *i.e.*, the computer's performance of logical operations in a way that produces the desired practical consequence. One cannot compare, therefore, the underlying processes of a computer program with, say, the underlying plot structure of a novel or a screenplay of a movie. This, of course, is the distinction recognized by the Supreme Court long ago in the seminal decision of *Baker v. Selden*.").

excerpts from Lotus video, copies of which are on file in the clerk's office).

Lotus nevertheless argues that if "method of operation" is used in this patent-law sense, then by analogy the computer program would also be the "method of operation" of the computer, a result inconsistent with Congress' intent to protect computer programs. Lotus Br. at 43. In fact, however, at the time of the enactment of the Copyright Act in 1976, it appeared that source code and object code might fall in the gap between patent and copyright, and not be protected by *either* regime. In closing this gap, the copyright laws appeared well-suited to the task. On the one hand, as the CONTU Report makes amply clear, the industry's concern was with software piracy – the literal copying of a disk and sale of the copy for a cheaper price. *See, e.g.*, CONTU Report at 10-11, 22-23. Copyright was well-suited to protect against such literal copying: the model developed to prohibit the unauthorized duplication of books and sound recordings could readily be applied to the unauthorized duplication of floppy disks. *See id.* at 10-11. On the other hand, such protection did not appear to pose an unduly chilling effect on later programmers, because there were virtually an "infinite" number of ways of writing programs to achieve the same result, "comparable to the theoretically infinite number of ways of writing Hamlet." *Id.* at 20 n.106.

Lotus now proposes to turn this legislation on its head, and obtain a copyright on the one thing that Congress was expressly assured would remain free. The CONTU Report advised Congress that, under the proposed legislation, one was "always free to make the machine do the same thing as it would if it had the copyrighted work placed in it," so long as this result was obtained "by one's own creative effort rather than by piracy." *Id.* at 21. Indeed, the Report went so far as to advise Congress that, if its use was "necessary to achieve a certain result," a later programmer could even *literally* copy the program's code. *Id.* at 20. Accordingly, so long as the later programmer used its own code, "one is always free to make [the computer] perform any conceivable process (in the absence of a patent)." *Id.*

It is precisely such "certain results" that Lotus now seeks to block by copyright. As has already been pointed out, Lotus does not contend that Borland copied Lotus' source code or object code,

or the internal structure of the two programs, at *any* level of abstraction. Indeed, Lotus did not even put its own code into evidence, a striking testament, one can presume, to the complete dissimilarity of the two programs. Instead, Borland looked at something *external* to the program – its "certain results" – and developed its own program to achieve those results.

Lotus' proposed monopoly on features external to the program has dramatically different economic consequences than the protection on code contemplated by Congress. One of the unique features of computer programs is that they may be built "on top" of one another, a feature that has been central to the use of computer programs to turn one machine (a computer) into an almost unlimited number of other "virtual" machines (a typewriter; telephone; calculator; and so forth). For programs to work in this way, it is necessary that the programs be able to "plug into" one another, using what are called a program's "interface specification."[61]

Allowing a monopoly on a computer program's "interface specification," such as the Lotus menu command hierarchy, would give the copyright holder a windfall, not only because it profits from users' investment in learning the particular command hierarchy (and resulting reluctance to switch), but because users have invested in developing macro programs "on top of" Lotus' own program. As Judge Boudin succinctly summarized, "it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus." *Lotus v. Borland* at 821, Pet.App. 26a-27a.

A rule allowing a copyright monopoly on the program's "socket" threatens to give the owner enormous market leverage not

---

61 *See* U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (Washington, D.C.: Government Printing Office, May 1992) at 126: "Programs have an external design or interface – the conventions for communication between the program and the user or other programs. The external design is conceptually separate from the program code that implements the interface (the internal design). It specifies the interactions between the program and the user or other programs, but not how the program does the required computations. There are typically many different ways of writing a program to implement the same interface."

just in its market – as here, where Lotus can block Borland from creating a product that connects to users' programs – but in other markets as well. An external interface is not unique or even unusual among computer programs, but rather is common at every level, as "applications" programs (such as spreadsheets) connect to "operating systems" programs (such as Windows), and so forth. Indeed, the entire stand-alone software industry ultimately depends upon the ability of such programs to be "compatible" with – *i.e.*, match the external structure of – the programs that run the computer hardware itself. It is for that reason that all participants in the software industry – users, developers, industry groups, computer scientists – are agreed that Lotus' position, if adopted, would have potentially calamitous consequences for the software industry in the United States.

Such results are a far cry from the limited protection of source and object code enacted by Congress. Menu command hierarchies and computer programs are not the same. As a matter of economics, copyright protection for source and object code blocks only one of a virtually unlimited number of ways of achieving the same "certain results"; copyright protection for these "certain results," such as a menu command hierarchy, potentially creates vast market power likely to chill further innovation. As a matter of Congressional intent, Congress plainly sought to provide copyright protection for source and object code; and, equally plainly, it sought to leave their "processes" and "methods" – including such "methods of operation" as a menu command hierarchy – free for all to use in the absence of a patent.

Congress' intent to exclude matter such as Lotus' menu command hierarchy seems plain. Even if it were ambiguous, however, this Court repeatedly has exhorted, as noted earlier, that the copyright laws must be interpreted in light of their basic purpose, which is to serve the public good. *Twentieth Century Music*, 422 U.S. at 156; *Sony*, 464 U.S. at 432. As the Constitution itself has made clear, intellectual property rights are not free, but are imposed at the expense of the public itself; what may be withdrawn from the public domain for the enjoyment of private monopoly, therefore, has been carefully limited and circumscribed, both under patent and copyright.

If Congress had not spoken, the Court has expressed its reluctance to extend the scope of such monopolies. Here, however, Congress has spoken, at every turn: it has made clear that the copyright laws are not to be used to undermine the integrity of the patent system; and it has equally explicitly made clear that copyright protection for computer programs is to be interpreted narrowly, and consistent with that intent. Lotus' menu command hierarchy is not a "computer program" for purposes of Section 102(a); even if it were an element of a program under Section 102(a), however, it would fall within the scope of matter excluded under Section 102(b). The Court of Appeals' decision is squarely in accord with this Congressional intent, and leaves copyright law in harmony with its Constitutional mandate of promoting the public good – a harmony badly set out of key by Lotus' proposed monopoly. It is accordingly respectfully submitted that the decision should be affirmed.

Respectfully submitted,

GARY L. REBACK
(Counsel of Record)
MICHAEL BARCLAY
SUSAN A. CREIGHTON
COLLEEN BAL
LISA A. DAVIS
WILSON, SONSINI, GOODRICH & ROSATI
Professional Corporation
650 Page Mill Road
Palo Alto, California 94304
Telephone: (415) 493-9300

ROBERT H. KOHN
BORLAND INTERNATIONAL, INC.
100 Borland Way
Scotts Valley, California 95066
Telephone: (408) 431-1801

PETER E. GELHAAR
KATHERINE L. PARKS
DONNELLY, CONROY & GELHAAR
One Post Office Square
33rd Floor
Boston, Massachusetts 02109
Telephone: (617) 542-7400

Attorneys for Respondent

December 1995

APPENDIX

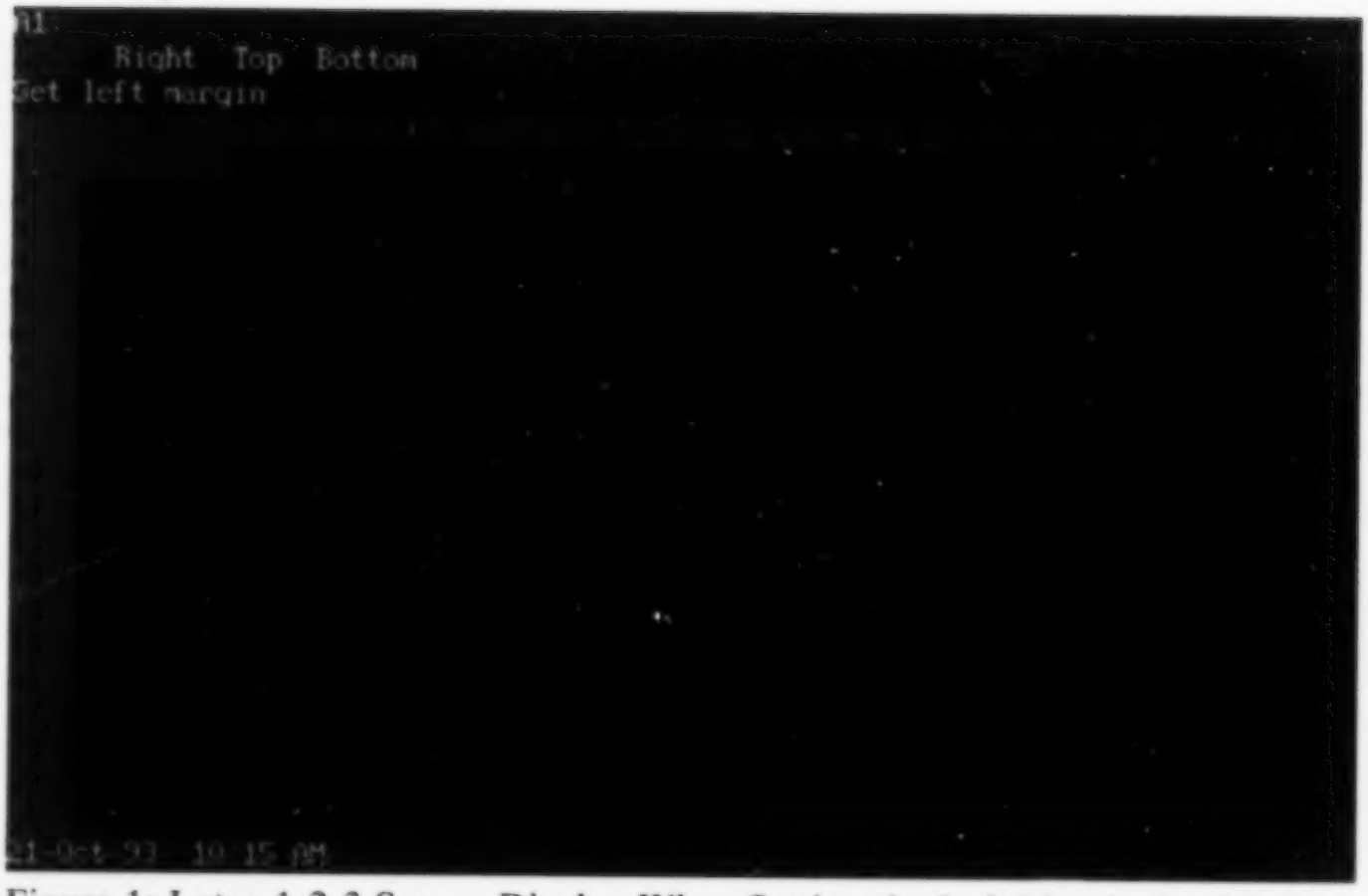


Figure 1: Lotus 1-2-3 Screen Display When Setting the Left Margin for Printing

A-1



Figure 2: Borland's Quattro Pro Screen Display in "123-Compatible Mode" When Setting the Left Margin for Printing

A-2

BEST AVAILABLE COPY

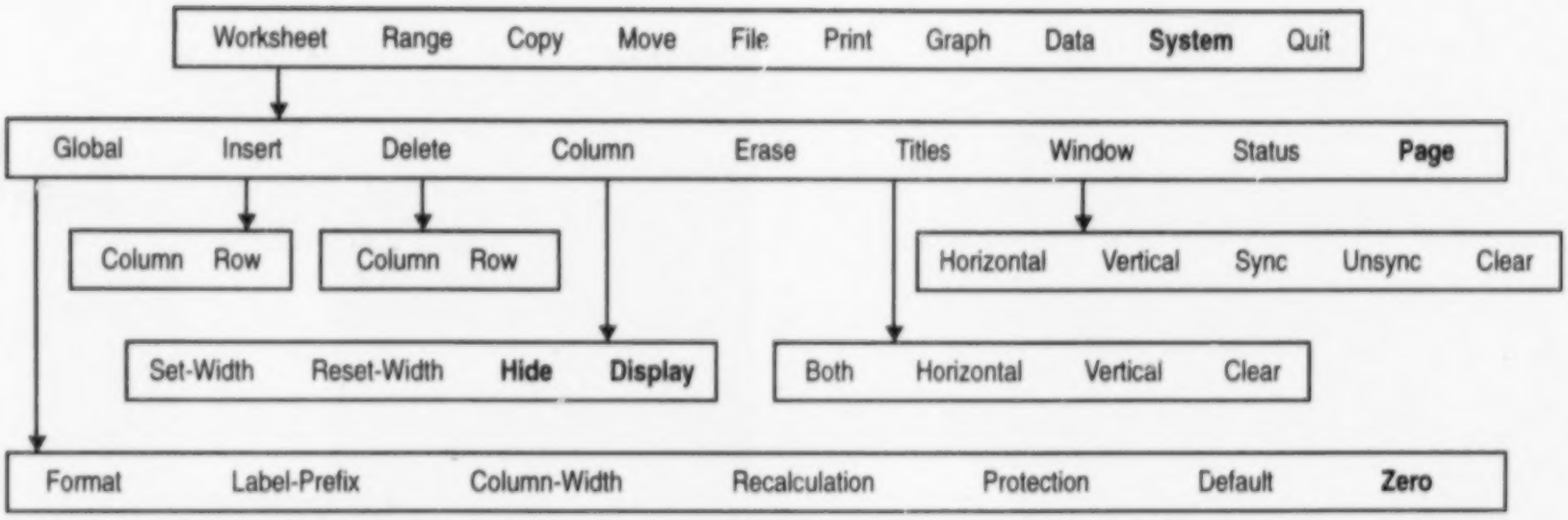


Figure 3: Lotus Command Tree

BEST AVAILABLE COPY

This page intentionally left blank.



Figure 4: Lotus 1-2-3 Screen Display, Showing Available Command Choices and Long Prompt Message

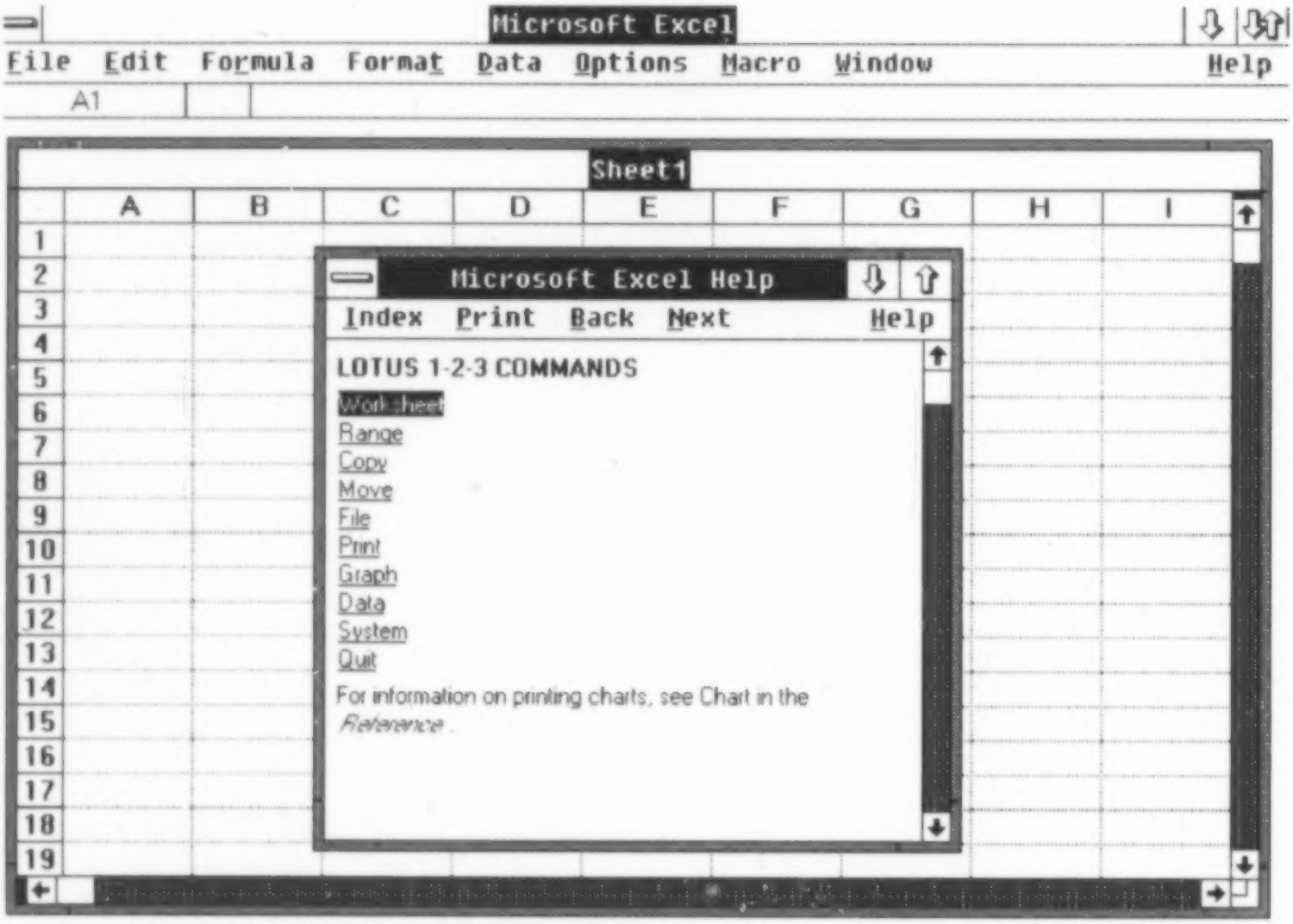


Figure 5: Microsoft's Excel Screen Display, with Lotus Menus, cited by Judge Keeton as Non-Infringing in the Paperback case

No. 94-2003

Supreme Court, U.S.
FILED
DEC 28 1995
CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

REPLY BRIEF FOR THE PETITIONER

NEAL D. GOLDMAN
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-1127

DONALD J. ROSENBERG
INTERNATIONAL BUSINESS MACHINES CORPORATION
44 South Broadway
White Plains, New York 10601
(914) 288-4010

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

STATEMENT PURSUANT TO RULE 29.6

See pp. i-iv of Petitioner's Reply In Support of Petition for a Writ of Certiorari.

TABLE OF CONTENTS

	PAGE
STATEMENT PURSUANT TO RULE 29.6	i
TABLE OF AUTHORITIES	iii
INTRODUCTION	1
I. BORLAND MISSTATES THE CASE IN CRITICAL RESPECTS	2
A. Borland Mischaracterizes the Record	3
B. The Issue is Expression	7
II. PATENT VS. COPYRIGHT—A FALSE DICHOTOMY	8
III. BORLAND FAILS TO ANSWER LOTUS' SECTION 102(b) ARGUMENT	12
IV. BORLAND'S AMICI URGE POLICY ARGUMENTS BETTER DIRECTED TO CONGRESS	16
CONCLUSION	20

TABLE OF AUTHORITIES

Cases	PAGE
Aro Mfg. Co. v. Convertible Top Replacement Co., 365 U.S. 336 (1961)	8n
Baker v. Selden, 101 U.S. 99 (1879)	10, 12-13, 17n
Bleistein v. Donaldson Lithographing Co., 188 U.S. 239 (1903)	16n
Blunt v. Patten, 3 F. Cas. 763 (C.C.S.D.N.Y. 1828)	20
Concrete Machinery Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600 (1st Cir. 1988)	4n
Emerson v. Davies, 8 F. Cas. 615 (C.C.D. Mass. 1845)	4n
Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340 (1991)	1n, 4, 11n, 12
Harper & Row, Publishers, Inc. v. Nation Enter., 471 U.S. 539 (1985)	6
Icicle Seafoods, Inc. v. Worthington, 475 U.S. 709 (1986)	5n
In re Lowry, 32 F.3d 1579 (Fed. Cir. 1994)	9n
Ker v. State of California, 374 U.S. 23 (1963)	3
Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470 (1974)	10n, 11n
Lipton v. The Nature Co., No. 94-9123, 1995 WL 700377 (2d Cir. Nov. 28, 1995)	7n
Mazer v. Stein, 347 U.S. 201 (1954)	10, 10n

PAGE

Roemer v. Bd. of Public Works of Maryland, 426 U.S. 736 (1976) 3n

Stewart v. Abend, 495 U.S. 207 (1990) 19n

Taylor Instrument Cos. v. Fawley-Brost Co., 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944) 10n

Time, Inc. v. Firestone, 424 U.S. 448 (1976) 3n

Statutes

17 U.S.C. § 101 ..... 2, 13n, 14, 14n
§ 102(a) 14, 15
§ 102(b) ..... *passim*
§ 108 ..... 17n
§ 110 ..... 17n
§ 111 ..... 17n
§ 115 ..... 17n
§ 119 ..... 18n

28 U.S.C. § 1295(a)(1) 11n

35 U.S.C. § 101 ..... 9n
§ 112 ..... 8n

Legislative Materials

H.R. Rep. No. 1476, 94th Cong., 2d Sess., *reprinted in* 1976 U.S.C.C.A.N. 5659 ..... 14, 14n

Rules and Regulations

Sup. Ct. R. 24.5 ..... 3n

Proposed Examination Guidelines for Computer-Implemented Inventions, 60 Fed. Reg. 28,778 (June 2, 1995) 9

PAGE

U.S. Patent and Trademark Office, "Legal Analysis to Support Proposed Examination Guidelines for Computer-Implemented Inventions," October 3, 1995 ..... 9n

Other Authorities

Paul Andrews, "User Friendly," *Seattle Times*, Aug. 31, 1993 ..... 6n

James M. Burger, "Fair Fight for Software," *Wash. Post*, Jan. 22, 1994 ..... 18n

Anthony L. Clapes, *et al.*, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. Rev. 1493 (1987) 14n

Anthony L. Clapes, *Softwars: The Legal Battles for Control of the Global Software Industry* (1993) .. 18n

Final Report of the National Commission on New Technological Uses of Copyrighted Works (the "CONTU Report") (1978) 13, 13n

Melville B. Nimmer and David Nimmer, *Nimmer on Copyright*, (1995) § 2.19 ..... 10n
§ 13.03[B] 4n

Andrew Pollack, "U.S. Protesting Japan's Plan to Revise Software Protection," *N.Y. Times*, Nov. 22, 1993 ..... 18n

T.R. Reid and Peter Behr, "A Software Fight's Blurred Battle Lines; U.S. Computer Companies Are on Both Sides as Japan Considers Copyright Law Changes," *Wash. Post*, Jan. 11, 1994 ..... 18n

PAGE

Pamela Samuelson, Randall Davis, Mitchell D. Kapor & Jerome H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308 (1994) 17n

Random House College Dictionary (rev. ed. 1988).... 7

U.S. Patent and Trademark Office and U.S. Copyright Office, "Patent-Copyright Laws Overlap Study," submitted July 17, 1991 ..... 9n

This case is not about jet aircraft cockpits, wall sockets, bicycle pedals, or any other of the increasingly far-fetched analogies that Borland and its *amici* proffer in place of reasoned principle to try to justify the First Circuit's decision. Rather, this case concerns an original selection and arrangement of *words*, displayed in a structured order and sequence in the menus of the 1-2-3 computer program. Lotus never sought, and the district court never granted, copyright protection for the functionality that the 1-2-3 program provides. The issue always has been whether a competitor may copy the arrangement of words Lotus chose to *express* the program's functionality to users, in order to offer a substitute for the original work.

The First Circuit held that Lotus' original arrangement of words is an unprotected "method of operation" pursuant to Section 102(b) of the Copyright Act of 1976 (the "1976 Act" or "Copyright Act"), 17 U.S.C. § 102(b), because its purpose is to assist users in working with the program—*i.e.*, because the menus, *inter alia*, tell users which sequences of keys to strike in order to achieve certain results when using 1-2-3. The First Circuit reached this extreme conclusion as a matter of law, without disturbing the district court's factual findings that 1-2-3's authors could have chosen innumerable other selections and arrangements of words to accomplish the same purpose, and that the particular choices they made were neither necessary to, nor dictated by, the functionality that the program provides. Borland makes no real effort to defend this unprecedented misinterpretation of Section 102(b). Borland fails to cite a single case—from any court—that has similarly construed Section 102(b) as denying protection to expression because of its utilitarian or "functional" *content*.[1]

---

[1] Indeed, except for a passing reference to this Court's decision in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991), Borland fails to cite *any* case decided under Section 102(b). Thus, it completely ignores the body of case law built up over the past fifteen years in which the courts have applied that provision to user interfaces and other aspects of computer programs. *See* Lotus Br. at 37-39.

Instead, Borland evades the issue by claiming that this case is governed not by copyright law, but by patent law. Borland Br. at 2, 19, 24. Borland contends that Lotus cannot prevent others from copying the precise arrangement of words in the 1-2-3 menus under copyright because any such protection falls within the province of utility patents. *Id.* at 21, 29-30, 44. This novel argument rests upon a fundamental misunderstanding of patent law and of the relationship between patents and copyrights. Borland's premise that utility patents cover words that convey information is incorrect. And its impractical suggestion that copyrightability determinations under Section 102(b) should be made by negative inference from the patent law, rather than through traditional copyright analysis (*id.* at 29, 31), finds no support in the text or legislative history of the 1976 Act.

Ultimately, Borland proffers nothing more than new definitional short-cuts to replace the First Circuit's. Like the First Circuit, it fails to appreciate the thrust of Section 102(b): that a work may both embody an unprotected "idea" or "method" *and* contain copyrightable expression in the author's particular statement of that "idea" or "method." When—as the district court found in this case—separable expression exists, that expression is entitled to copyright protection. Neither the patent law, nor the definition of a "computer program" set forth in Section 101 of the Copyright Act, nor anything else Borland cites can provide a defensible basis for the First Circuit's holding to the contrary.

I

BORLAND MISSTATES THE CASE IN CRITICAL RESPECTS

The case Borland describes to the Court bears little resemblance to the one the parties actually tried. This case was tried under copyright law, not patent or antitrust law, and both the proof at trial and the findings of the district court were limited to establishing the elements of a copyright infringement claim.

A. Borland Mischaracterizes the Record

Borland pretends that anything ever filed in the district court is established as record fact, whether or not it was offered in evidence or was the subject of fact-finding at trial. For example, of the 22 exhibits Borland cites in its Brief, 18 were not proffered at trial.[2] Borland asks this Court to accept these unproven assertions as "fact" despite the Court's repeated admonitions that it does not "sit as in nisi prius to appraise contradictory factual questions." *Ker v. State of California*, 374 U.S. 23, 34 (1963).[3] We focus on those issues where Borland's presentation is most misleading.

1. *Originality of 1-2-3.* Borland suggests that the 1-2-3 user interface is not novel because certain of its stylistic elements (such as the use of a two-line moving cursor menu) could be found in earlier programs written by 1-2-3's author, and because individual words found in 1-2-3's menus had appeared in previous works. Borland Br. at 3-4. These contentions never were tried below because Borland did not

---

[2] Borland has failed to heed Rule 24.5 of this Court, which requires that any exhibit referred to in a brief on the merits shall indicate where in the record that exhibit "was offered in evidence" and "ruled on by the judge." *See, e.g.*, Borland Br. at 11 n.19 (citing to list of summary judgment submissions at JA 78-79). The videotapes Borland lodged with the Court are a good example. Only a small portion of Borland's original video was admitted at trial, and then only for a limited purpose not relevant here. Dkt. No. 405 at 3-6. Borland's reply video, which Lotus moved to strike when submitted for summary judgment consideration (Dkt. No. 131 at 9-10), was never offered at trial.

[3] *See also Roemer v. Bd. of Public Works of Maryland*, 426 U.S. 736, 758 (1976) (refusing to reappraise factual findings based upon party's "gleaning" of "occasional evidence" from a voluminous record); *Time, Inc. v. Firestone*, 424 U.S. 448, 464 (1976) (declining to "canvass the record" to make an initial determination of fact not found by lower court).

challenge the originality of the 1-2-3 menus, taken as a whole (*see Feist*, 499 U.S. at 356-58; Borland Br. at 2-3; Dkt. No. 66 at 14), and anything else was irrelevant to the adjudication of Lotus' claims.[4]

2. *Quality of the parties' products.* Borland attempts to disparage the role that the 1-2-3 menus played in the product's commercial success and boasts that its infringing products were inherently superior. Borland Br. at 11-12. Again, Borland relies upon "evidence" that never was offered at trial, in support of contentions upon which the district court never was asked to rule. To determine Borland's liability for taking the 1-2-3 menus, the district court was required to find merely that Borland copied a "substantial part" of the infringed work.[5] Having ruled that the 1-2-3 menus easily met this test, the district court did not need to determine their importance relative to other aspects of the program. Nor did it have any reason to determine the truth of Borland's claims of product superiority. Pet. App. 83a-84a, 140a, 180a. An infringer cannot avoid liability merely by claiming that it has improved upon, or made additions to, the original.[6]

---

[4] As Justice Story wrote 150 years ago in *Emerson v. Davies*, 8 F. Cas. 615, 619 (C.C.D. Mass. 1845): "No man creates a new language for himself, at least if he be a wise man, in writing a book. He contents himself with the use of language already known and used and understood by others . . . . He, who constructs by a new plan, and arrangement, and combination of old materials, in a book designed for instruction . . . has a title to a copy-right, which cannot be displaced by showing that some part of his plan, or arrangement or combination, has been used before."

[5] Pet. App. 49a, 73a, 84a, 119a-120a, 152a-153a. *See Concrete Machinery Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 608 (1st Cir. 1988) (holding that the taking of more than "trivial aspects of another's work" constitutes infringement).

[6] *See* 3 Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* § 13.03[B] (1995) (hereinafter "*Nimmer*") at 13-61-62 ("If substantial similarity is found, the defendant will not be immunized from liability by reason of the addition in his work of different characters or additional and varied incidents, nor generally by reason of his work proving more attractive or saleable than the plaintiff's.").

3. *Competitive and market impact of Borland's copying.* Borland rides a second horse in the opposite direction by contending that, despite the alleged "superiority" of Quattro, it could not compete with Lotus in 1987 unless it copied the 1-2-3 menus.[7] The First Circuit apparently agreed. Its decision assumes that the popularity of 1-2-3 created a "lock-in" effect that precluded the introduction of superior products that did not copy the 1-2-3 menus (Pet. App. 20a, 26a), and that this market condition continued to prevail as late as 1995. *Id.* at 26a. What Borland omits, and the First Circuit ignored, is that this issue was not litigated in the district court and therefore was not raised properly on appeal.[8] The extent to which copying the 1-2-3 menus contributed to the sales of Quattro and Quattro Pro was to have been the subject of the damages trial, which did not occur.[9] That the First Circuit could base its decision upon an erroneous assumption concerning Lotus' alleged market dominance, years after 1-2-3 had been eclipsed

---

[7] Borland Br. at 13 n.23. Borland's sole supporting citation is the 1991 deposition testimony of its chairman, Philippe Kahn. At that deposition, however, Mr. Kahn also testified:

> Q. Is it still your belief that in order for Quattro or Quattro Pro to be marketable it needs to execute 1-2-3 macros?
>
> A. I don't know. (JA 657)

[8] *See Icicle Seafoods, Inc. v. Worthington*, 475 U.S. 709, 714 (1986) (vacating judgment because appellate court "should not simply have made factual findings on its own").

[9] By the time the First Circuit's decision mooted the damages trial, Borland had indicated its intent to call expert witnesses to opine that Borland's copying had, at most, a modest effect upon its sales. These included the economist William Baumol of Princeton and N.Y.U., who supervised econometric analyses purporting to show *no* measurable "network effect" attributable to the 1-2-3 menus, and the marketing researcher John Hauser of M.I.T., who conducted a survey of Borland's customers. *See* Dkt. No. 624, Ex. B at 3 (Baumol); Dkt. No. 652, Ex. A (Hauser). Although Lotus had intended to offer evidence demonstrating that Borland's copying had a larger effect on its sales, the issue was never resolved.

by Microsoft Excel, is a perfect illustration of the danger in decision-making unsupported by record fact.[10]

More fundamentally, Borland did not assert an antitrust counterclaim or similar defense in the district court. Thus, typical antitrust issues such as market definition, barriers to entry, or market power (due to "network effects" or other factors) were not tried below. This was, and remains, a copyright case; it has *never* been a principle of copyright law that the reward for a work's popularity is forfeiture of the owner's property rights. As the Court held in *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 539, 559 (1985), it is "fundamentally at odds with the scheme of copyright to accord lesser rights in those works that are of greatest importance to the public." The First Circuit's introduction of "competitive impact" analysis in determining the copyrightability of a work is both unprecedented and unwarranted. Indeed, if this Court were to adopt the view that an antitrust-like inquiry should be made before a copyright can be held valid, the result would be a dramatic increase in the burden and expense of copyright infringement litigation in a manner Congress never intended.

---

[10] Borland attempts to explain Excel's market success, in part, by claiming that Excel provided 1-2-3 compatibility while Borland was enjoined from doing so. Borland Br. at 16. This explanation is both outside the record and contrary to historical fact. By the time Borland was enjoined in August 1993, sales of Excel already had surpassed those of 1-2-3 and Quattro Pro, as can be verified from publicly available information. *See* Paul Andrews, "User Friendly," *Seattle Times*, Aug. 31, 1993, at D3. And, although Excel did attempt to ease the transition for former 1-2-3 users, it did so through features unlike Borland's "1-2-3 emulation" interface, including the "Help for 1-2-3 Users" screen Borland includes in its Brief (at A-5) to suggest, falsely, that Excel also had "emulation" menus. *See* Pet. App. 39a, 237a. The district court was never called upon to determine (i) whether anything in Excel infringed Lotus' copyrights, or (ii) why Excel achieved market success. Borland invites the Court to join the First Circuit in substituting speculation for a trial record.

B. The Issue is Expression

Contrary to Borland's misstatement of the district court's holding as extending protection to a "selection and arrangement of executable operations" (Borland Br. at 1, 8, 18, 43), the district court held that a program could provide exactly the *same* functional operations that 1-2-3 provides without infringing Lotus' copyrights, so long as it used different words to express those operations to users. Pet. App. at 86a-89a, 128a-133a. The court protected only the coupling of the "selection and arrangement" of operations with their *expression* in the 1-2-3 menus—like the arrangement of unprotected facts in a copyrighted compilation. Pet. App. 40a-41a, 132a-133a. Borland's "native" menus duplicated the functions of 1-2-3, but Lotus never claimed that they infringed. The district court found that Borland's "1-2-3 emulation" menus infringed because Borland had copied 1-2-3's expression, not its selection of functions.

Indeed, Borland does not deny that the 1-2-3 menus contain expression separable from the program's functionality. Borland concedes that the words in the 1-2-3 menus are not buttons, but a "depiction" of the various steps a user may take to instruct the program to perform various operations. Borland Br. at 5. To "depict" means "to represent or characterize in words; describe." *Random House College Dictionary* 357 (rev. ed. 1988). And while Borland claims to discern a "firm line" between the words in the menus and other explanatory aspects of the 1-2-3 product, such as a narrative description in the user's manual or a few lines of text in its on-screen "help" facility (Borland Br. at 6-7), Borland concedes that even an individual long prompt describing an individual menu command—such as "Copy a cell or range of cells"—may qualify for copyright protection. *Id.* Borland cites nothing to suggest that the difference between collections of one-word descriptions and seven-word descriptions is meaningful for copyright purposes,[11] or that an

---

[11] *See Lipton v. The Nature Co.*, No. 94-9123, 1995 WL 700377, at *4-*6 (2d Cir. Nov. 28, 1995) (upholding copyright protection in par-

abbreviated reference guide—such as a summary or index—cannot constitute expression merely because a *more* detailed explanation is contained elsewhere in the work.

II

PATENT VS. COPYRIGHT—A FALSE DICHOTOMY

Borland erects a classic "straw man" argument when it asks the Court to decide "whether copyright extends to the subject matter of utility patents." Borland Br. at 44. Although Borland tries to portray Section 102(b) as establishing two mutually exclusive worlds of intellectual property protection (*id.* at 29), as if the Copyright Act were nothing more than the "Not-Patent Act," it fails to demonstrate that any genuine conflict between the two bodies of law *does* exist with regard to the subject matter of this case.

The essential premise of Borland's argument is that the arrangement of words in the 1-2-3 menus might have been protected by a utility patent, and therefore may not be protected by copyright. But Borland is consistently vague as to the form this hypothetical patent would take. What, exactly, is the claim it imagines could be drafted to cover a particular arrangement of words, but *only* that arrangement of words?[12] Borland ignores the long-standing "printed matter" doctrine of patent law, which provides that "claims defining as the invention certain novel arrangements of printed lines or char-

ticular selection and arrangement of 77 three-word "animal terms of venery" such as a "leap of leopards").

12 Borland misperceives basic principles of patent interpretation when it asserts that disclosure of the "best mode" pursuant to 35 U.S.C. § 112 in a patent claiming a process or method of menu display, comparable to those referenced in Borland's Brief, "would doubtless cover the words themselves." Borland Br. at 35. It is well settled that the grant of a patent covers only the claims, and does not extend to the specifications, descriptions, or embodiments of the claimed invention that the applicant is required to disclose. *Aro Mfg. Co. v. Convertible Top Replacement Co.*, 365 U.S. 336, 339 (1961). None of the patents Borland cites, in fact, claims a particular arrangement of words.

acters, useful and intelligible only to the human mind" are not patentable.[13] Under the PTO's Proposed Examination Guidelines for Computer-Implemented Inventions, 60 Fed. Reg. 28,778 (June 2, 1995), patents still will not issue for claims that define "information rather than a computer-implemented process or specific machine."[14] Because the purpose of the 1-2-3 menus is to provide "information," Borland's assumption that they could be the subject of a utility patent is simply incorrect.

Borland's contention that certain terms in Section 102(b) should be construed as incorporating principles of patent law (Borland Br. at 31-35) is both wrong and irrelevant. Nothing in the text or legislative history of the 1976 Act suggests that Congress intended the Patent Act and the Copyright Act to be read as a single, integrated statute, with common definitions and cross-referencing provisions.[15] Nor does Borland explain how Congress *could* have intended the two statutes to be read together with regard to the copyrightability of computer programs, in light of the fact (which Borland does not dispute) that computer programs, standing alone, were not considered patentable when Congress passed the 1976 Act and the 1980 Software Amendments.

13 *In re Lowry*, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (citation omitted). When asked by Congress to address the subject of user interface patentability in 1991, the Patent and Trademark Office ("PTO") declared that "[t]he mere display on a [computer] screen of commands, menus, questions and answers, forms, or icons is not generally considered patentable subject matter for utility patents" under the "printed matter" doctrine. U.S. Patent and Trademark Office and U.S. Copyright Office, "Patent-Copyright Laws Overlap Study," submitted July 17, 1991 (the "Joint Study"), at 45-46.

14 U.S. Patent and Trademark Office, "Legal Analysis to Support Proposed Examination Guidelines for Computer-Implemented Inventions," Oct. 3, 1995, at III.B.1.

15 Only one term—"process"—is found in both Section 102(b) of the Copyright Act and Section 101 of the Patent Act (35 U.S.C. § 101, defining statutory subject matter).

The point is irrelevant because patent and copyright protection may coexist in the same work if the separate eligibility requirements are met.[16] A patent may be obtained for a novel "process," even as a copyright will protect an original expression of that "process," whether or not it is patented. *Baker v. Selden*, 101 U.S. 99, 102-03 (1879). The two statutes reward different forms of creativity in different ways, but their policies do not conflict.[17] As this Court declared in *Mazer v. Stein*, 347 U.S. 201, 217 (1954) (footnote omitted):

> Neither the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted. We should not so hold.

Borland contends that the principle enunciated in *Mazer* is limited to design patents (Borland Br. at 30), but no court has so held and Borland fails to present any rational reason why this should be so.[18] Thus, even if Borland is correct that the terms of Section 102(b) should be read as excluding patentable subject matter, the Court still must address the

[16] In a joint submission to the House Subcommittee on Intellectual Property and the Administration of Justice, the PTO and the Copyright Office reported in 1991 that a computer program could both comprise part of a patented process *and* be protected by copyright. Joint Study at 10-11. *See also* 1 *Nimmer on Copyright*, § 2.19 at 2-204.30 ("It will seem on principle that if a work otherwise meets the requirements of copyrightability, it should not be denied such, simply because the claimant happens to be entitled to supplementary protection under other Legislation.").

[17] *See Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 491 (1974) (stating that concurrent trade secret protection for patentable inventions does not conflict with the policies behind the Patent Act).

[18] Borland's flawed reading of *Mazer* rests upon a footnote in which the Court merely referred to a party's citation of *Taylor Instrument Cos. v. Fawley-Brost Co.*, 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944), to support the party's contention that utility patents and copyrights could not "overlap." 347 U.S. at 215 n.33. *Taylor* itself held only that copyright could not be obtained for a chart that "neither teaches nor explains the use of the art" (139 F.2d at 100)—a factual finding contrary to the district court's in this case regarding the 1-2-3 menus.

"idea/expression" (or "patent/expression," in Borland's view) dichotomy codified in that provision. Nothing in the patent law can assist in that task.

Borland's approach would also dramatically complicate the adjudication of even simple copyright cases. Under its theory, every infringement case involving copyright in a utilitarian work also would become a patent case, in which the district court would first need to determine patentability in the abstract and without the benefit of an examined patent claim.[19] Moreover, this new body of "patentability" decisions, rendered in cases arising under the Copyright Act, would sabotage the uniform development of patent law that the Federal Circuit was created to provide.[20]

Ultimately, Borland's repeated refrain that Lotus is improperly seeking to obtain "patent-like protection" through copyright (*e.g.*, Borland Br. at 21) can be dismissed as mere rhetoric. The district court never held that Lotus' copyrights gave it a "patent-like" monopoly on the use of menus to express the functionality of a spreadsheet program. If it had, Borland's "native" menus also would have infringed, despite their use of different words. Rather, the district court extended protection only to the original selection and arrangement of words Lotus chose to employ, and found infringement due to Borland's copying of that selection and arrangement. Pet. App. 36a, 84a. Far from "patent-like," this is the traditional protection afforded by copyright.[21]

[19] *See Kewanee*, 416 U.S. at 491 (recognizing the "almost impossible burden" of a rule requiring courts to determine patentability without an examined claim before granting trade secret protection).

[20] Because cases raising only copyright claims do not fall within the jurisdiction of the Federal Circuit pursuant to 28 U.S.C. § 1295(a)(1), appellate review of these "shadow" patent rulings would lie in each of the twelve other courts of appeals.

[21] Indeed, because Judge Keeton found that Borland included "virtually identical" copies of the 1-2-3 menus in its products (*id.*), the menus would be protected even under the "thin" copyright protection afforded to compilations of facts or data. *See Feist*, 499 U.S. at 349.

III

BORLAND FAILS TO ANSWER LOTUS' SECTION 102(b) ARGUMENT

Borland flatly misstates Lotus' argument concerning the proper interpretation of Section 102(b). *See* Lotus Br. at 26-29. Lotus does not argue, as Borland suggests, that the only operative term in Section 102(b) is the word "idea," and that the remaining terms are either extraneous or redundant. Borland Br. at 31. Rather, our argument, properly understood, is that each of those terms stands in exactly the same relationship to protected "expression" as does an "idea." Section 102(b) establishes an overarching distinction between the original expression of a particular subject and the subject itself. That is, just as there is an "idea/expression" dichotomy, there is also a "system/expression," a "method/expression," a "process/expression"—and even a "fact/expression" (*see Feist*, 499 U.S. at 350)—dichotomy. In each case, the task of the court is the same—to undertake a factual inquiry to determine whether the work at issue contains identifiably separable "expression." If so, that expression is entitled to protection. If not, then the expression and the "idea" (or "method" or "process") have merged, and the work is unprotected.

The district court faithfully undertook this task, finding after two trials that the expression in the 1-2-3 menus could be separated from their underlying functionality (or "system" or "method"). Under its ruling, anyone is free to publish a spreadsheet program that performs the same functions, uses the same kind of menus, and even organizes its menus in the same way according to the same principles. All that is prevented is the use in another spreadsheet program of the same *words* in the same order to express the same functions. The court's decision flows directly from *Baker v. Selden*:

> The use by another of the same methods of statement, whether in words or illustrations, in a book published for teaching the art, would undoubtedly be an infringement of the copyright. (101 U.S. at 104.)

The First Circuit did not reject the district court's findings on this subject as clearly erroneous; it simply ignored them as "immaterial" to its analysis. Pet. App. 17a. Other than drawing an inapposite analogy to VCR buttons (*id.* at 18a-20a), the First Circuit never even attempted to explain why those words were not conceptually distinguishable from the functions they represented. Neither does Borland.

Instead, Borland contends that the menus are unprotected because they do not fit the statutory definition of a "computer program," based upon references in the CONTU Report to "source and object code."[22] CONTU never stated, however, that copyright protection would be limited as Borland suggests,[23] and it would be error to ignore the plain meaning of the statutory language even if CONTU had.[24] Nor does

---

[22] Borland Br. at 47. This is why Borland so vigorously resists any suggestion that the words in menus appear in a program's code. *Id.* at 11. Yet Borland's own description of the "iterative" process by which the 1-2-3 menus were developed, stressing the extent to which each change in the menus required subsequent implementation in the code (*id.* at 5), proves that Lotus' assertions regarding 1-2-3 are correct. And Borland cannot deny that these same words appear in its code as well: the developer of Quattro Pro agreed at deposition that "somewhere in the code" of the program every word appearing in the menus was "actually typed out." JA 764-65.

[23] For example, CONTU anticipated copyright protection for flowcharts, computer databases, and many other computer-based works other than source and object code. CONTU Report at 1, 16, 21, 25 & 38.

[24] As the First Circuit implicitly acknowledged, the 1-2-3 menus *do* fit within the express terms of the definition. *Compare* Pet. App. 27a (the purpose of the menus "is to transmit directions from the user to the computer, *i.e.*, to *operate* the computer"; emphasis in original) *with* 17 U.S.C. § 101 (a "computer program" is "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result").

Borland adequately address the fact that the 1-2-3 menus also fit comfortably within the Section 101 definition of a "literary work." Borland Br. at 45-46. This fact cannot be so easily dismissed, however, because Congress explicitly stated its intention for computer programs to receive protection under the 1976 Act as "literary works" *before* it received CONTU's recommendations.[25]

But even if we assume, *arguendo*, that the 1-2-3 menus fit within neither of these statutory definitions, Borland still is mistaken in arguing that they cannot be protected by copyright. Borland misperceives the role that the Section 101 definitions play in the statutory scheme. They neither limit the grant of copyright in Section 102(a) nor specify the scope of protection Congress has afforded to copyrightable works.[26] As the legislative history makes clear, the categories of protected works identified in Section 102(a) and defined in Section 101 were intended to be "illustrative, not limitative." House Report at 53, *reprinted in* 1976 U.S.C.C.A.N. at 5666. One simply cannot reconcile the expansive language of Section 102(a), which expressly provides that copyright protects original works of authorship fixed in any medium of expression "now known or later developed," with Borland's stunted view that Congress intended copyright protection for computer programs to provide only "limited protection of source and object code." Borland Br. at 49.

---

[25] *See* H.R. Rep. No. 1476, 94th Cong., 2d Sess., at 53 (the "House Report"), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5666.

[26] For example, although the definition of a "motion picture" is a "series of related images which, when shown in succession, impart an impression of motion, together with accompanying sounds, if any" (17 U.S.C. § 101), no court has interpreted that definition as precluding protection for a motion picture's plot, dialogue, or characters—all of which are plainly protected. *See* Declaration of Melville B. Nimmer, ¶24, *reprinted in* Anthony L. Clapes, *et al.*, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. Rev. 1493, 1591 (1987).

Thus, whether the 1-2-3 menus can be characterized as a "computer program" or "literary work," more simply as a "work of authorship," or as a non-literal element of a computer program (as the district court did, Pet. App. 126a-28a, 224a-26a), ultimately is irrelevant. Under any or all of these labels, they comprise copyrightable subject matter under Section 102(a). The question then is whether Section 102(b) denies them protection because their purpose is to describe and explain a particular set of functional operations. Neither Borland nor the First Circuit has provided any reasoned principle this Court could employ to reach that result.

Simply put, the words in the 1-2-3 menus are not buttons on any machine. They do not act or cause the computer to act. They serve to inform or remind the user as to which keystrokes to strike in a particular order to cause the program to perform a particular operation. The user does not type or speak the words found in the menus—the program would not respond to either such form of input.[27] Rather, the user either strikes certain lettered keys on the keyboard, or moves the cursor and hits the "Enter" key, in order to select commands and communicate those selections to the program. Even when commands are selected by use of the cursor, the program does not respond to the word itself, but to the position of the cursor on the screen when the "Enter" key is hit. The program would do exactly the same things at exactly the same times if the words were blanked out on the computer screen or, indeed, if the monitor was turned off.[28] The only difference would be that the user, deprived of the information those words provide, would have to guess or remember which keys to strike in order to bring about the desired result.

---

[27] Borland's assertion that commands "may also be selected when they are spoken into a computer equipped with voice recognition software" (Borland Br. at 3) is pure invention. Whatever software Borland is referring to, it is not part of 1-2-3.

[28] This point can be confirmed by reference to the "Software Library" included in the record transmitted to this Court, which contains all the relevant programs installed on a computer. Dkt. No. 360.

If this informative function is sufficient to render the 1-2-3 menus an inseparable part of a method of operation, then the same is true of any sheet of piano music. Like 1-2-3's menus, the sheet music contains a series of notations that tell the pianist which keys to strike in which order. Like the 1-2-3 user, as the pianist practices and learns a particular composition, he or she becomes less dependent upon the visual assistance provided by the music (menus), until some or all of the keystrokes are memorized. As with spreadsheet programs, each new musical composition requires the pianist to learn new keystrokes. The First Circuit labelled this "absurd" in the context of a computer program (Pet. App. at 20a), but presumably would have no objection if the subject were Mozart instead. The First Circuit's difficulties in understanding or appreciating the expression in the 1-2-3 menus is no excuse, however, for its failure to protect that expression in accordance with Congress' mandate.[29]

IV

BORLAND'S AMICI URGE POLICY ARGUMENTS BETTER DIRECTED TO CONGRESS

Borland's *amici* (more than half of whom acknowledge being subsidized by Borland) present a varied assortment of policy reasons why, in their view, the Lotus 1-2-3 menus ought not to be protected by copyright, but fail to demonstrate through competent legal argument that the menus are *not* protected under current law.[30] The short answer to their con-

[29] *See Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 251 (1903) (Holmes, J.) (reversing decision that advertisements were *per se* ineligible for copyright protection, and warning that "[i]t would be a dangerous undertaking for persons trained only to the law" to attempt to judge the worth of copyrightable works, "outside of the narrowest and most obvious limits").

[30] For example, the "Copyright Law Professors" alone offer seven conflicting theories in support of the outcome—if not the reasoning—of

tentions is that they are lobbying the wrong branch of government. If they believe that Congress gave too little weight to the policy concerns they advance or that circumstances have changed since Congress passed the 1976 Act in ways that Congress did not (or could not) anticipate, their remedy is to attempt to persuade Congress to amend the statute.[31]

Instead, Borland's *amici* advocate that this Court should establish new limitations on the scope of copyright protection for computer programs—or, in the First Circuit's words, "devis[e] a new doctrine that Borland's use is privileged." Pet. App. 27a. But the Copyright Act is already replete with numerous exceptions, including detailed regulatory schemes involving compulsory licensing and royalties, intended to address those circumstances that Congress determined should be covered by special rules.[32] None of these exceptions

the First Circuit's decision. Brief *Amicus Curiae* of Copyright Law Professors, at 3-4. Incredibly, this group of copyright academicians argues, upon the authority of *Baker v. Selden*, that copyright in "functional writings" does not extend to the expression of "functional content" (*id.* at 14-15)—despite the Court's explicit statement in *Baker* that copyright will protect an author's original description or explanation of a patentable "mixture" or of a useful "art." *Baker v. Selden*, 101 U.S. at 102-03, 104. They do not even try to reconcile their assertion with Congress' decision to recognize copyright protection for computer programs in full awareness of their functional purpose. If a line of source code such as "ADD A,B" is not an expression of "functional content," it is difficult to imagine what is.

[31] Indeed, counsel of record for the Copyright Law Professors has argued that neither copyright nor patent law appropriately protects computer programs and proposes, instead, a new *sui generis* intellectual property regime, under which the 1-2-3 menus would receive protection against "clones" and "near-clones" like Borland's products for a limited period of time. *See* Pamela Samuelson, Randall Davis, Mitchell D. Kapor & Jerome H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308, 2312-14 (1994) (stating that such protection is necessary to prevent "clones" from destroying the "incentives to invest in software innovation").

[32] *See, e.g.*, 17 U.S.C. § 108 (permitting limited reproduction by libraries and archives); § 110 (permitting instructional performances in classrooms); § 111 (establishing compulsory licensing of television cable operators); § 115 (establishing compulsory licensing for phonorecords);

applies to this case. If Congress agreed that user interfaces of computer programs are simply too important to protect (or should only enjoy protection with special limitations on scope or duration),[33] or that users require greater protection for their investments in training or macros,[34] or that success sufficient to generate some unspecified level of "network effects" ought to divest copyright owners of their preexisting property rights,[35] Congress is perfectly capable of drafting a statutory provision to address any and all of these concerns. To date, Congress has declined to do so.

There are good reasons why Congress has not seen fit to adopt the approach of the *amici*. For example, the "network effects" argument boils down to a contention that the popularity of a work results in the forfeiture of its copyrights—a point of view that has never before been credited in American

§ 119 (establishing compulsory licensing scheme regarding certain satellite transmissions).

33 *See, e.g.*, Brief *Amicus Curiae* of American Committee for Interoperable Systems ("ACIS"), at 3, 20 (arguing for exception where copying is intended to promote program "interoperability"). The Court should hesitate to assume that ACIS (or the other Borland *amici*) fairly represent the industry on whose behalf they purport to speak. ACIS, for example, was formed specifically for the purpose of advocating its members' interest in maximizing their ability to develop products for "computer systems developed by other companies." *Id.* at 1. Its worldwide campaign in service of this goal has frequently brought its views into conflict with others in the industry as well as the United States government. *See, e.g.*, Andrew Pollack, "U.S. Protesting Japan's Plan to Revise Software Protection," *N.Y. Times*, Nov. 22, 1993, at D2; T.R. Reid and Peter Behr, "A Software Fight's Blurred Battle Lines; U.S. Computer Companies Are on Both Sides as Japan Considers Copyright Law Changes," *Wash. Post*, Jan. 11, 1994, at D1; James M. Burger, "Fair Fight for Software," *Wash. Post*, Jan. 22, 1994, at A16. *See generally* Anthony L. Clapes, *Softwars: The Legal Battles for Control of the Global Software Industry* 8-9 (1993) (describing the global struggle between "software originators and software copiers").

34 Brief *Amicus Curiae* of Users Groups, at 16-19. Of course, the users would have a broad "fair use" defense, unavailable to Borland.

35 Brief *Amicus Curiae* of Economics Professors and Scholars, at 15.

copyright law.[36] Moreover, many of the concerns the *amici* raise are not unique to computer programs at all, but are inherent in copyright protection generally. "Network effects" in one form or another surround many types of traditionally copyrightable works.[37] Similarly, the complaint of the "Users Groups" that their ability to exploit their learning or the macros they have written for use with Lotus 1-2-3 may be limited, to some extent, if others are not free to copy that program,[38] is scarcely unusual under copyright law. Indeed, "switching costs" are an ordinary fact of economic life, whether or not copyrights are involved. The inconvenience of learning a new spreadsheet program or re-writing macros is not materially different from that experienced by a repertory theater group (or a major motion picture studio) that must abandon a long-planned production because it has lost its rights to the underlying work. And all these difficulties pale in comparison to the predicament of the sailors whose ability to avoid the Nantucket shoals depended upon access to a

36 Indeed, the entire discussion about the value of the 1-2-3 menus being derived from the investment of others is a red herring. That the award-winning, classic motion picture "Rear Window" may have reflected vastly more creative value and financial investment than the obscure magazine article that inspired it did not excuse the filmmaker's failure to secure the necessary rights from the author of the article. *See Stewart v. Abend*, 499 U.S. 207, 211-13, 227-28 (1990).

37 For example, teachers may develop extensive course materials, tests, and plans of instruction centered around a "standard" copyrighted textbook. No one would argue that this gives the teachers a right to duplicate the textbook if the publisher issues a materially revised edition, or gives another publisher a right to satisfy the teachers' pent-up demand by copying the original textbook. Nor does the existence of extensive markets for toys, clothes, and similar items relating to popular movies or imaginary characters like "Barbie" or "Snoopy" imply that the rights in the works at the hub of such a "network" are in any way diminished.

38 Brief *Amicus Curiae* of Users Groups, at 20-21. The Users Groups concede that the "barrier" to adoption of a new spreadsheet program "is not an insuperable one." *Id.* at 16. The closest they come to giving a quantified answer to their own question concerning "how much time would be required for retraining" is "a lot." *Id.*

copyrighted chart. *Blunt v. Patten*, 3 F. Cas. 763 (C.C.S.D.N.Y. 1828). Yet never before has a copyright been lost because of such concerns.

Viewed from a consumer's short-term perspective, copyright protection is always a bother—it limits the free availability of the works we wish to see, hear, and use, and no doubt requires us to pay more to obtain them. The Copyright Act, however, reflects Congress' judgment—and the judgment embodied in the Constitution—that providing these incentives to authors will enhance the public welfare in the long run, by inspiring a plentiful supply of new and original works. At bottom, Borland and its *amici* ask the Court to strike a different balance in this case, but can point to nothing in the 1976 Act or its legislative history to justify that result.

CONCLUSION

The decision of the First Circuit should be reversed and the case remanded for further proceedings.

Respectfully submitted,

NEAL D. GOLDMAN
LOTUS DEVELOPMENT CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-1127

DONALD J. ROSENBERG
INTERNATIONAL BUSINESS MACHINES CORPORATION
44 South Broadway
White Plains, New York 10601
(914) 288-4010

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
JEFFREY E. OSTROW
LORI E. LESSER
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

Counsel for Petitioner

December 28, 1995

...OTION FILED
NOV 9 1995

(10)

No. 94-2003

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE AND BRIEF AMICUS CURIAE OF AMERICAN INTELLECTUAL PROPERTY LAW ASSOCIATION IN SUPPORT OF NEITHER PARTY

DON W. MARTENS *
President
AMERICAN INTELLECTUAL PROPERTY LAW ASSOCIATION
2001 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 415-0780

CHARLES L. GHOLZ
OBLON, SPIVAK, MCCLELLAND, MAIER & NEUSTADT, P.C.
Suite 400
1755 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 413-3000

BAILA H. CELEDONIA
ROBERT W. CLARIDA
COWAN, LIEBOWITZ & LATMAN, P.C.
1133 Avenue of the Americas
New York, NY 10036-6799
(212) 790-9200

Counsel for Amicus Curiae

* Counsel of Record

WILSON - EPES PRINTING CO., INC. - 789-0096 - WASHINGTON, D.C. 20001

BEST AVAILABLE COPY

21 pp

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE

The American Intellectual Property Law Association ("AIPLA"), a national association with interests primarily in the areas of copyright, patent, trademark, trade secret and other intellectual property, respectfully moves for leave to file the attached brief *amicus curiae* in this copyright infringement case. The consent of the attorney for petitioner has been obtained. The consent of the attorney for respondent was requested but refused.

In 1980, Congress adopted the Final Report of the National Commission on New Technological Uses of Copyrighted Works ("CONTU Report") which recommended that Congress continue to protect computer programs as literary works under copyright law. CONTU Report 11 (1978). AIPLA makes this motion because the First Circuit's reasoning, underlying its decision to find petitioner's computer program menu command hierarchy to be

per se uncopyrightable, ignores this congressional mandate and abandons the fundamental Constitutional analysis heretofore used to distinguish between protected, copyrightable expression and unprotected idea.

AIPLA is in a unique position as an *amicus* because it does not offer the views of a partisan party, or of an industry or segment thereof. Rather, AIPLA's interest is in developing and maintaining a body of law which is consistent with the purposes of the Constitution's Copyright Clause.

AIPLA therefore respectfully requests that the Court grant its motion for leave to file the accompanying brief, and consider AIPLA's arguments during its deliberations.

Respectfully submitted,

DON W. MARTENS *
President
AMERICAN INTELLECTUAL
PROPERTY LAW ASSOCIATION
2001 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 415-0780

CHARLES L. GHOLZ
OBLON, SPIVAK, MCCLELLAND,
MAIER & NEUSTADT, P.C.
Suite 400
1755 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 413-3000

BAILA H. CELEDONIA
ROBERT W. CLARIDA
COWAN, LIEBOWITZ & LATMAN, P.C.
1133 Avenue of the Americas
New York, NY 10036-6799
(212) 790-9200

Counsel for Amicus Curiae

November 9, 1995

* Counsel of Record

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
INTEREST OF AMICUS CURIAE	1
SUMMARY OF ARGUMENT	2
ARGUMENT AND CITATIONS OF AUTHORITY	3
I. THE COPYRIGHT ACT PROTECTS COMPUTER PROGRAMS AS LITERARY WORKS	3
II. THE FIRST CIRCUIT MISUNDERSTANDS THE PURPOSE AND MEANING OF SECTION 102(b)	6
A. The First Circuit Took a Dangerous and Improper Shortcut in Applying § 102(b), Ignoring the Explicit Language of the Copyright Act	7
B. The First Circuit Approach Ignores the Intent of Congress as Reflected in the Legislative History	8
C. The First Circuit Approach Ignores the Judicial Consensus as to the Meaning of § 102(b)	9
D. The First Circuit Approach Misinterprets *Baker v. Selden* and the Merger Doctrine	11
III. THE UNINTENDED RESULT OF THE FIRST CIRCUIT'S EXPANSIVE READING OF § 102(b) COULD ADVERSELY AFFECT COPYRIGHT LAW GENERALLY	12
CONCLUSION	14

TABLE OF AUTHORITIES

CASES — Page

Apple Computer, Inc. v. Formula International Inc., 725 F.2d 521 (9th Cir. 1984) 9

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) 4, 5, 9

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995) 10

Autoskill, Inc. v. National Educational Support Systems, 994 F.2d 1476 (10th Cir.), *cert. denied*, 114 S. Ct. 307 (1993) 9, 12

Baker v. Selden, 101 U.S. 99 (1869) 7, 11, 12

Cable/Home Communication Corp. v. Network Productions, Inc., 902 F.2d 829 (11th Cir. 1990) 10

CCC Information Services, Inc. v. Maclean Hunter Market Reports, Inc., 44 F.3d 61 (2d Cir. 1994), *cert. denied*, 116 S. Ct. 72 (1995) 10

Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 4, 12

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994) 11, 12

Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340 (1991) 7

Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823 (10th Cir. 1993) 4

Harper & Row, Publishers, Inc. v. Nation Enterprises, 471 U.S. 539 (1985) 7

Johnson Controls, Inc. v. Phoenix Control Systems, Inc., 886 F.2d 1173 (9th Cir. 1989) 4

Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527 (5th Cir.), *cert. denied*, 115 S. Ct. 82 (1994) 4, 10

Lotus Development Corp. v. Borland International Inc., 49 F.3d 807 (1st Cir. 1995) ..... *passim*

M. Kramer Manufacturing Co. v. Andrews, 783 F.2d 421 (4th Cir. 1986) 10

Mazer v. Stein, 347 U.S. 201 (1954) 7, 11

TABLE OF AUTHORITIES—Continued

Page

Micro-Sparc, Inc. v. Amtype Corp., 592 F. Supp. 33 (D. Mass. 1984) 5

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931) 4, 7

Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487 (2d Cir. 1960) 7

Ricker v. General Electric Co., 162 F.2d 141 (2d Cir. 1947) 7

Robert R. Jones Associates, Inc. v. Nino Homes, 858 F.2d 274 (6th Cir. 1988) 10

Toro Co. v. R & R Products Co., 787 F.2d 1208 (8th Cir. 1986) 10

Walker v. Time-Life Films, Inc., 784 F.2d 44 (2d Cir.), *cert. denied*, 476 U.S. 1159 (1986) 7

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 4, 10

STATUTES

17 U.S.C. § 101 ..... 3, 13

17 U.S.C. § 102(b) ..... *passim*

17 U.S.C. § 117 4

Computer Software Amendments of 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980) 5

General Agreement on Tariffs and Trade, Uruguay Round Implementation Act, Pub. L. No. 103-826 (Dec. 8, 1994) 5

North American Free Trade Agreement, Pub. L. No. 130-182 (Dec. 8, 1993) 5

MISCELLANEOUS

Council Directive on the Legal Protection of Computer Programs, 91/250/EEC, art. 6, 1991 O.J. (L. 122) 42 ..... 5, 6

Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 COLUM. L. REV. 2559, 2570 (1994) 13

TABLE OF AUTHORITIES—Continued

	Page
H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667	3, 4
David Nimmer & Melville B. Nimmer, 1 NIMMER ON COPYRIGHT § 1.10[B][2] (1995)	6
William F. Patry, I COPYRIGHT LAW & PRACTICE 319 (1994)	6

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF AMICUS CURIAE OF AMERICAN INTELLECTUAL PROPERTY LAW ASSOCIATION IN SUPPORT OF NEITHER PARTY

INTEREST OF AMICUS CURIAE

American Intellectual Property Law Association ("AIPLA") files this *amicus curiae* brief to aid the Court in its review of the ruling of the United States Court of Appeals for the First Circuit in the above-titled case.

AIPLA is a national association of more than 9,000 members, primarily attorneys, whose interests and practices lie in the areas of copyright, patent, trademark, trade secret and other intellectual property law. Membership in AIPLA is drawn from private law firms, corporations, universities and government. Attorney members of AIPLA represent both copyright owners and their competitors.

AIPLA does not support either party to this appeal nor have an interest in the outcome of this appeal, except to the extent that the outcome affects copyright law, thereby affecting AIPLA members and the public generally. Rather, AIPLA offers the views of a leading organization of intellectual property attorneys representing a broad range of views. Accordingly, AIPLA's concerns that the First Circuit's reasoning violates basic principles of both copyright law and statutory interpretation are the product of a non-partisan analysis.

SUMMARY OF ARGUMENT

The United States Court of Appeals for the First Circuit applied Section 102(b) of the Copyright Act, 17 U.S.C. § 102(b),[1] to deny copyright protection for the expression it assumed was contained in the menu command hierarchy of the Lotus 1-2-3 spreadsheet program. *Lotus Development Corp. v. Borland International Inc.*, 49 F.3d 807, 815 (1st Cir. 1995) ("*Lotus*").

The First Circuit accepted the finding of the district court that the Lotus 1-2-3 menu command hierarchy contained expressive elements, but it concluded that the entirety of the menu command hierarchy was a "method of operation" under Section 102(b) and thus did not constitute copyrightable subject matter. *Lotus*, 49 F.3d at 815-16. Defining "method of operation" as "the means by which a person operates something," *id.* at 815, the First Circuit determined that, because the Lotus command hierarchy is necessary to operate the Lotus program, it is therefore unprotectible. The First Circuit's approach is an unnecessary departure from the analysis

[1] 17 U.S.C. § 102(b) provides:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

employed by other courts, and it ignores fundamental principles of copyright law, codified by Congress in the Copyright Act.

AIPLA expresses no view as to whether the Lotus 1-2-3 menu command hierarchy is idea or expression, whether the hierarchy constitutes expression which has merged with the idea it embodies, whether any other copyright doctrine may require filtering out of certain expression, nor whether any other defenses are applicable. AIPLA believes, however, that the reasoning used by the First Circuit to reach its ultimate result is incorrect.

ARGUMENT

I. THE COPYRIGHT ACT PROTECTS COMPUTER PROGRAMS AS LITERARY WORKS

Under § 101 of the Copyright Act, a computer program is defined as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. That same section of the Act defines "literary works" as:

> [W]orks, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

Id. Computer programs are therefore literary works by statutory definition.[2]

[2] The legislative history confirms that Congress intended to encompass computer programs within the definition of "literary works" in the Copyright Act:

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also includes computer data bases, and *computer programs* to the extent that they incorporate

Apart from the specific exceptions set forth in § 117 of the Act,[3] Congres did not create a lesser or different standard of protection for computer programs than for any other literary works. Thus, just as the plot, characters, and other non-literal elements of a novel or play may constitute protectible authorship, *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931), courts have recognized the copyrightability of the non-literal elements of computer programs.[4]

The legislative history of the Copyright Act supports this widespread judicial recognition of copyrightable au-

---

authorship in the programmer's expression of ideas, as distinguished from the ideas themselves.

H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667 ("House Report") (emphasis added).

[3] 17 U.S.C. § 117 permits the owner of a copy of a computer program to make an additional copy of the program for archival purposes, or where the making of such a copy is "an essential step in the utilization of the computer program in conjunction with a machine and . . . is used for no other purpose." The § 117 exceptions to the copyright owner's exclusive rights clearly contemplate that computer programs are to be afforded copyright protection. *See Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1248 (3d Cir. 1983) ("the language of [§ 117] . . . by carving out an exception to the normal proscriptions against copying, clearly indicates that programs are copyrightable and are otherwise afforded copyright protection"), *cert. dismissed*, 464 U.S. 1033 (1984) (*"Apple Computer"*).

[4] *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir. 1992) (*"Altai"*) ("[I]f the non-literal structures of literary works are protected by copyright; and if computer programs are literary works, as we are told by the legislature; then the non-literal structures of computer programs are protected by copyright."). *See also Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1248 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 536 n.20 (5th Cir.), *cert. denied*, 115 S. Ct. 82 (1994); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1177 (9th Cir. 1989); *Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9 F.3d 823, 840 (10th Cir. 1993).

thorship in the non-literal aspects of computer programs. *See* House Report at 54. When Congress amended the Copyright Act in 1980,[5] it adopted the definition of "computer program" recommended in 1978 by the National Commision on New Technological Uses of Copyrighted Works ("CONTU"), thereby reconfirming that computer programs are to be protected under the same fundamental principles as other literary works.[6]

This approach to the protection of computer programs as literary works has been confirmed more recently in the international context, as Congress has ratified the North American Free Trade Agreement ("NAFTA") and adopted legislation to implement its provisions in the United States. Pub. L. No. 130-182 (Dec. 8, 1993). Among the intellectual property provisions of NAFTA is the commitment of member countries to protect computer programs under copyright law as literary works. NAFTA, Art. 1705(1)(A).

Similarly, the Agreement on Trade-Related Intellectual Property Rights ("TRIPS") which the U.S. adopted in 1994 in connection with the General Agreement on Tariffs and Trade ("GATT"),[7] requires member nations to protect computer programs as literary works. TRIPS, Art. 10(1). In Europe, the 1991 Council Directive on the Legal Protection of Computer Programs, 91/250/

---

[5] Computer Software Amendments of 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980).

[6] *See Final Report of the National Commission on New Technological Uses of Copyrighted Works* 11-14 (1978) ("CONTU Final Report"). Because Congress adopted CONTU's recommended amendments in 1980 almost verbatim and without discussion, courts have looked to the CONTU Final Report as an accurate reflection of Congressional intent. *See, e.g., Apple Computer*, 714 F.2d at 1252; *Micro-Sparc, Inc. v. Amtype Corp.*, 592 F. Supp. 33, 35 n.7 (D. Mass. 1984).

[7] Uruguay Round Implementation Act, Pub. L. No. 103-826 (Dec. 8, 1994).

EEC, art. 6, 1991 O.J. (L. 122) 42, also mandates literary work protection for computer programs. Thus, Congress has led the world in recognizing that computer programs must be protected in the same manner and to the same extent as other original works of authorship.

II. THE FIRST CIRCUIT MISUNDERSTANDS THE PURPOSE AND MEANING OF § 102(b)

Copyright protection for a computer program, as for any "original work of authorship," extends only to its expression, and not to any "idea . . . process, system [or] method of operation . . . embodied in such work." 17 U.S.C. § 102(b). This section of the Copyright Law is a restatement of the idea/expression dichotomy, which has always been central to balancing the competing interests of copyright owners and the public by keeping ideas in the public domain while encouraging authors to create new works.[8] Melville B. Nimmer & David Nimmer, 1 NIMMER ON COPYRIGHT § 1.10[B][2](1995); William F. Patry, 1 COPYRIGHT LAW AND PRACTICE 319 (1994).

By interpreting § 102(b) as an independent, *a priori* bar to copyright protection for expressive works of authorship—in this case, the expression of a method of operation—the First Circuit ignores the balance inherent in the idea/expression dichotomy. This interpretation of § 102(b) is also inconsistent with the explicit language of the Copyright Act, the intent of Congress as reflected in the legislative history, and the clear consensus of the other appellate courts which have addressed the issue. Further, the First Circuit's approach reflects a misunderstanding of the

[8] The legislative history confirms that Congress intended to restate, and not change the law in this regard.

> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged.

House Report at 57.

holding of this Court in *Baker v. Selden*, 101 U.S. 99 (1869) ("*Baker*"), which is consistent with the idea/expression dichotomy.

A. The First Circuit Took a Dangerous and Improper Shortcut in Applying § 102(b), Ignoring the Explicit Language of the Copyright Act

For works of authorship of all kinds, in order to determine whether infringement of copyrightable material has occurred, a court must separate protectible expression from the unprotectible ideas contained in a work. *See, e.g., Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 347 (1991) (telephone directories); *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 547 (1985) (presidential memoirs); *Mazer v. Stein*, 347 U.S. 201, 217 (1954) (ornamental lamps); *Walker v. Time-Life Films, Inc.*, 784 F.2d 44, 49 (2d Cir.), *cert. denied*, 476 U.S. 1159 (1986) (fact-based motion picture); *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 488 (2d Cir. 1960) (fabric designs); *Ricker v. General Electric Co.*, 162 F.2d 141, 144-46 (2d Cir. 1947) (book containing scientific materials); *Nichols*, 45 F.2d at 121 (play).

The First Circuit, however, expressly refused to separate the protectible expression which it presumed to exist in the Lotus menu command hierarchy, either by applying the abstraction-filtration-comparison methodology articulated by the Second Circuit in *Altai*, or by applying any other form of analysis. Rather, it chose a simplistic and novel view of § 102(b). It saw no need to separate the unprotectible ideas embodied in the Lotus menu command hierarchy from the specific concrete expression which the district court found Lotus had created to embody those ideas. *Lotus*, 49 F.3d at 815-16.[9] Instead, the First Circuit concluded that, because "users operate Lotus 1-2-3 by using the menu command hierarchy," the expression

[9] The court accepted the district court's holding that the command hierarchy was in fact expressive. *Lotus*, 49 F.3d at 815-16.

contained in the Lotus hierarchy is part of the "method of operation" of Lotus 1-2-3 and is thus excluded from copyright protection under Section 102(b) of the Copyright Act. *Id.* at 816.

B. The First Circuit Approach Ignores the Intent of Congress as Reflected in the Legislative History

Contrary to the First Circuit's position, Congress intended § 102(b) merely to be a codification of the idea-expression dichotomy, not an alternative to it. The 1976 House Report directly addressed this issue in connection with the copyrightability of computer programs:

> Section 102(b) is intended, among other things, to make clear that the *expression* adopted by the programmer is the copyrightable element in a computer program, and that the actual *processes* or *methods* embodied in the program are not within the scope of the copyright law. Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the *basic dichotomy* between expression and idea remains unchanged.

House Report, at 57 (emphasis added).[10]

The CONTU Final Report similarly confirmed the centrality of the idea-expression distinction in the computer context, noting that the "words of a program" are copyrightable even when they "are used ultimately in the implementation of a process." CONTU Final Report at 21. Moreover, the CONTU Report directly expressed the Congressional understanding that copyright in a computer program is not confined to the literal code, but extends to more abstract, non-literal elements as well, provided that they constitute expression, rather than "idea" or "process":

> The line which must be drawn [to distinguish protectible from unprotectible elements of a program] is between the *expression* and the *idea*, between the *writing* and the *process* which is described.

CONTU Final Report at 25 (emphasis added).

The First Circuit's holding that § 102(b) absolutely precludes copyright protection for concrete expression, such as the selection and arrangement of the Lotus menu hierarchy—which the court assumed to be expressive—without reference to the idea/expression dichotomy, *Lotus*, 49 F.3d at 816, is contradicted by the legislative history of the Copyright Act and the 1980 Computer Software Amendments.

C. The First Circuit Approach Ignores the Judicial Consensus as to the Meaning of § 102(b)

In light of the statutory text and the legislative history of § 102(b), other courts of appeals have uniformly rejected the argument that particular computer program elements are *a priori* a process, system, method of operation or idea without regard to whether they embody any expression. *See, e.g., Apple Computer*, 714 F.2d at 1250-51 (rejecting defendant's argument that operating system programs are "*per se* excluded from copyright" under § 102(b) as a "process" "system" or "method of operation"); *Apple Computer, Inc. v. Formula International Inc.*, 725 F.2d 521, 523 (9th Cir. 1984) (same); *Autoskill, Inc. v. National Educational Support Systems*, 994 F.2d 1476, 1493, 1495 n.23 (10th Cir.) ("*Autoskill*") (rejecting challenge that reading training program was "process", "method of operation" or "procedure" and explicitly holding that the abstraction-filtration-comparison analysis is appropriate to determine whether program keying procedure is protected or not), *cert. denied*, 114

[10] This passage from the House Report also demonstrates that Congress did not intend to draw fine distinctions among the various "exclusions" enumerated in § 102(b), such as "method," "process" and "idea." Particularly in the context of computer programs, the legislative history uses such terms interchangeably as antonyms for "expression."

S. Ct. 307 (1993); *M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 435-36 (4th Cir. 1986) (noting that courts reject that programs are *per se* a process or system and rejecting similar argument concerning computerized video game).

More generally, the courts of appeals have consistently interpreted § 102(b) as neither more nor less than a restatement of the idea-expression dichotomy. *See, e.g., CCC Information Services, Inc. v. Maclean Hunter Market Reports, Inc.*, 44 F.3d 61, 69 & n.12 (2d Cir. 1994), *cert. denied*, 116 S. Ct. 72 (1995); *Whelan*, 797 F.2d at 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); *Kepner-Tregoe*, 12 F.3d at 533 & n.8 (5th Cir.); *Robert R. Jones Associates, Inc. v. Nino Homes*, 858 F.2d 274, 277 (6th Cir. 1988); *Toro Co. v. R & R Products Co.*, 787 F.2d 1208, 1211-12 (8th Cir. 1986); *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995); *Cable/Home Communication Corp. v. Network Productions, Inc.*, 902 F.2d 829, 842-43 (11th Cir. 1990).

Accordingly, the First Circuit's ruling that § 102(b) automatically places expressive works of authorship outside the scope of copyright protection is not only unprecedented, it is contradicted by the overwhelming weight of authority, as well as by the statute and its legislative history. As the cases cited above recognize, a proper application of § 102(b) requires the court to decide as a threshold matter whether the authorship at issue constitutes idea or expression. If it falls on the "expression" side of that line, § 102(b) does not authorize a court to automatically deny copyrightability. Contrary to the First Circuit's characterization, *Lotus*, 49 F.3d at 815-16, this separation of idea and expression is not an optional innovation introduced by the Second Circuit's decision in *Altai*. It is a statutory *requirement* which the courts are not at liberty to disregard. Moreover, this is so whether

the work in question is a presidential memoir, an architectural plan, a film or, as here, computer software.

D. The First Circuit Approach Misinterprets *Baker v. Selden* and the Merger Doctrine

The sole support which the First Circuit cites for its conclusion that § 102(b) bars copyright in both expression and ideas is this Court's decision in *Baker*, cited at 49 F.3d 816-17. *Baker*, however, provides no support for the First Circuit's argument. It merely articulates the fundamental copyright doctrine of merger, whereby expression may be denied copyright protection if it is a "necessary incident" to the idea or process which it embodies. Thus, *Baker* does not eliminate the necessity of separating idea from expression, as the First Circuit erroneously concluded.

At issue in *Baker* were certain ruled ledger sheets for which the author of an accounting text sought copyright protection. This Court denied protection for the ledger sheets, not because the sheets themselves were an "idea" or "process", but because the court found that they must "necessarily be used as incident to" the system of accounting described in Baker's textbook. This Court reasoned that a copyright on the sheets would have extended impermissibly into a monopoly on the unprotectible system itself. It was for this reason the sheets were deemed uncopyrightable. Baker's textbook, however, did not lose its protection. Expression which merely describes a system, without foreclosing others from describing the same system in a different way, is protectible.

The courts consistently have held that *Baker* only stands for the proposition that expression merged with idea or process cannot be protected. However, expression which is *not* merged with an idea *is* protectible, even if it is part of a functional work. *Mazer*, 347 U.S. at 218. *See also Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1345 (5th Cir. 1994) (noting

that *Baker* held ledger sheets were not copyrightable because "they were necessary incidents" to the idea or process embodied in the bookkeeping system); *Autoskill*, 994 F.2d at 1493 (district court properly dealt with § 102(b) and the process/expression dichotomy as discussed in *Baker* by excluding the highest levels of abstraction and considering lower, more detailed levels in filtration analysis); *Altai*, 982 F.2d at 704-05 (noting that *Baker* held ledger sheets not copyrightable because they were "necessary incidents" to accounting systems, but *Baker* offered "scant guidance on how to separate idea or process from expression" or how to distinguish necessarily incident expression).

The other circuit courts which have dealt with the issue of the protectibility of the non-literal elements of a computer program have adopted some variation of the abstraction-filtration methodology, such as in the Second Circuit's *Altai* approach. As part of this analysis, once the idea and the expression have been separated in the abstraction step, the expression may be denied protection in the filtration step, *inter alia*, under *Baker* and the merger doctrine. Thus, *Baker* may preclude protection for certain particular expressions under certain circumstances, but it does not, under any analytic approach, authorize a court to invoke § 102(b) as an alternative to separating idea from expression, or automatically to deny copyright protection for expressive works.

III. THE UNINTENDED RESULT OF THE FIRST CIRCUIT'S EXPANSIVE READING OF § 102(b) COULD ADVERSELY AFFECT COPYRIGHT LAW GENERALLY

The First Circuit acknowledges that computer software code is generally copyrightable. *Lotus*, 49 F.3d at 816 n.11. Yet its definition of "method of operation" in § 102(b) ("a means by which a person operates something . . . ." *Id.* at 815), can be viewed as in conflict

with the statutory definition of a computer program as a "set of statements or instructions to be used directly or indirectly in a computer to bring about a certain result." 17 U.S.C. § 101.[11] Thus, there is a significant danger that the First Circuit's definition of method of operation will erode copyright protection for computer software far beyond menu command hiearchy.

The First Circuit's decision also could have unintended but pernicious implications beyond the field of computer software. A similar result could arise for the expressive elements of works considered by a court to be a "system" or "process". The broadbrush wording of the First Circuit's opinion does not suggest any limit to the scope of its view of § 102(b).

The First Circuit's interpretation of § 102(b), however, is unnecessary to maintain a balance between the competing interests in this case. A legitimate limit is already in place under existing copyright doctrine. It is the idea-expression dichotomy, codified with the passage of § 102(b), but dismissed by the First Circuit as unnecessary.

---

[11] A noted copyright scholar observed, "since we know that Congress did determine in 1980 to protect computer programs, the terms 'process,' 'system,' or 'method of operation' [found in § 102(b)] must not be understood literally." Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Softwear*, 94 COLUM. L. REV. 2559, 2570 (1994). Judge Boudin, in his concurring opinion in *Lotus*, frankly concedes that the apparent "string of exclusions" enumerated in § 102(b), including the term "method of operation" on which the majority relied, cannot be "taken literally" without excluding most computer programs from copyright protection. *Lotus*, 49 F.3d at 820. The First Circuit's overbroad understanding of "method of operation" must, therefore, be incorrect.

CONCLUSION

Amicus curiae AIPLA takes no position on the factual issues of this case or the issue of the copyrightability of the Lotus 1-2-3 menu command hiearchy. AIPLA urges only that the Court apply, in its analysis of that issue, fundamental principles of copyright law as set forth above.

Respectfully submitted,

DON W. MARTENS *
President
AMERICAN INTELLECTUAL
PROPERTY LAW ASSOCIATION
2001 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 415-0780

BAILA H. CELEDONIA
ROBERT W. CLARIDA
COWAN, LIEBOWITZ & LATMAN, P.C.
1133 Avenue of the Americas
New York, NY 10036-6799
(212) 790-9200

November 9, 1995
CHARLES L. GHOLZ
OBLON, SPIVAK, MCCLELLAND,
MAIER & NEUSTADT, P.C.
Suite 400
1755 Jefferson Davis Highway
Arlington, VA 22202-3586
(703) 413-3000

Counsel for Amicus Curiae

November 9, 1995

* Counsel of Record

Supreme Court, U.S.
FILED
NOV 9 1995
CLERK

No. 94-2003

IN THE
Supreme Court of the United States
OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

MOTION FOR LEAVE TO FILE AMICUS CURIAE BRIEF AND AMICUS CURIAE BRIEF OF DIGITAL EQUIPMENT CORPORATION, THE GATES RUBBER COMPANY, INTEL CORPORATION AND XEROX CORPORATION IN SUPPORT OF PETITIONER

MORTON DAVID GOLDBERG *
JUNE M. BESEK
DAVID O. CARSON
JESSE M. FEDER
SCHWAB GOLDBERG PRICE & DANNAY
1185 Avenue of the Americas
New York, NY 10036
(212) 575-8150
Attorneys for Amici Curiae
* Counsel of Record

WILSON - EPES PRINTING CO., INC. - 789-0096 - WASHINGTON, D.C. 20001

BEST AVAILABLE COPY

28 pp

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

MOTION FOR LEAVE TO FILE AMICUS CURIAE BRIEF

Digital Equipment Corporation, The Gates Rubber Company, Intel Corporation and Xerox Corporation ("Amici") respectfully move for leave to file the attached *amicus curiae* brief in this copyright infringement case. The consent of the attorney for petitioner has been obtained. The consent of the attorney for respondent was requested but refused.

Amici are leading developers of software and other computer products, which they create and market for mainframe, mini and personal computers for the business, education, government and entertainment markets. All of Amici develop computer software, to compete in the worldwide software market, to complement the products and services they provide their customers, or to support

other segments of their businesses. Amici constitute a major segment of the U.S. economy and, collectively, have annual revenues exceeding $44 billion, employ more than 192,000 people worldwide, and invest more than $3 billion annually in research and development.

Computer software development has become a key part of the United States economy. As with all copyright industries, it relies on appropriate levels of copyright protection for the works it creates. The decision below upsets this reliance, and undermines the industry's copyright incentives.

The First Circuit has denied copyright protection to expression in one of the most creative elements of a computer program—the interface between the computer and the user—because that element performs a useful function. The First Circuit's decision threatens to deny to any expressive program element that also performs a useful function the copyright protection conferred by Congress on authors of computer programs. It does this by interpreting Section 102(b) of the Copyright Act as a bar to protection of the expression of methods of operation.

This case is much more than a dispute between the parties: it is a clash between two fundamentally-opposed views of the protection Congress provided for computer software in the Copyright Act. That is why this case has attracted such extraordinary attention within the computer industry—as the ten *amicus curiae* briefs filed in the First Circuit attest.

One view embraces Congress' decision that computer programs are literary works, to be afforded protection under copyright law in accordance with the same principles that apply to all copyrighted works. In this view, appropriate protection under copyright law fosters innovative competition in the software industry and fulfills the Constitutional dictate "to promote the progress of science." U.S. Const., art. I, § 8, cl. 8. Amici hold this view, together with the greater part of the software industry.

The opposite view rejects Congress' decision that programs should be protected under copyright. By attempting to erect special judicial barriers to protection, those expressing this view seek to curtail the protection that the copyright statute confers.

As some of the leading innovators in the computer industry, Amici are in a position to bring to the Court's attention the viewpoint of a major portion of the business sector that is most affected by the decision below. Amici are in a better position than the parties to assist the Court in assessing the important implications of this case for industry and for the copyright system. The attached *amicus curiae* brief focuses on these considerations.

Amici respectfully request that this motion be granted and that the brief be considered by the Court in connection with its review of the judgment of the First Circuit.

Respectfully submitted,

MORTON DAVID GOLDBERG *
JUNE M. BESEK
DAVID O. CARSON
JESSE M. FEDER
SCHWAB GOLDBERG PRICE & DANNAY
1185 Avenue of the Americas
New York, NY 10036
(212) 575-8150
Attorneys for Amici Curiae

* Counsel of Record

November 9, 1995

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
INTEREST OF AMICI CURIAE	1
SUMMARY OF ARGUMENT	3
ARGUMENT	4
I. CONGRESS MANDATED THAT COPYRIGHT PROTECT THE EXPRESSION IN COMPUTER PROGRAMS, INCLUDING THE EXPRESSION OF METHODS OF OPERATION	6
A. Congress Determined that Computer Programs are Literary Works Protected by Copyright Under the Same Principles that Govern Protection of Other Literary Works	6
B. Section 102(b) of the Copyright Act Bars Protection Only for Ideas, Not for Expression of Ideas	10
II. IN HOLDING EXPRESSION OF A METHOD OF OPERATION TO BE UNCOPYRIGHTABLE PER SE, THE DECISION BELOW THWARTS CONGRESS' SCHEME FOR PROTECTING COMPUTER PROGRAMS	13
A. The First Circuit's Per Se Bar to Protecting User Interfaces Is a Misinterpretation of the Copyright Act	13
B. Denying Protection to Expression in Computer Programs Undermines the Incentives Congress Enacted for Their Authors	17
CONCLUSION	19

TABLE OF AUTHORITIES

Cases	Page
Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984)	18
Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, —— U.S. ——, 115 S. Ct. 1176 (1995)	12
Atari Games Corp. v. Oman, 888 F.2d 878 (D.C. Cir. 1989)	5, 16
Atari, Inc. v. North American Philips Consumer Elecs. Corp., 672 F.2d 607 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982)	12
Autoskill Inc. v. National Educ. Support Sys., Inc., 994 F.2d 1476 (10th Cir.), *cert. denied*, —— U.S. ——, 114 S. Ct. 307 (1993)	12
Baker v. Selden, 101 U.S. 99 (1879)	10
Blue Chip Stamps v. Manor Drug Stores, 421 U.S. 723 (1975)	13
Cable/Home Communication Corp. v. Network Products, Inc., 902 F.2d 829 (11th Cir. 1990)	12
Campbell v. Acuff-Rose Music, Inc., —— U.S. ——, 114 S. Ct. 1164 (1994)	8
CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc., 44 F.3d 61 (2d Cir. 1994), *cert. denied*, —— U.S. ——, 64 U.S.L.W. 3240 (1995)	12
Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992)	16
FDIC v. Meyer, 510 U.S. ——, 114 S. Ct. 996 (1994)	13
Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340 (1991)	8
Fogerty v. Fantasy, Inc., —— U.S. ——, 114 S. Ct. 1023 (1994)	8
Gates Rubber Co. v. Bando Chem. Indus., 9 F.3d 823 (10th Cir. 1993)	16
Gustafson v. Alloyd Co., —— U.S. ——, 115 S. Ct. 1061 (1995)	8
Harper & Row, Publishers v. Nation Enterprises, 471 U.S. 539 (1985)	19

TABLE OF AUTHORITIES—Continued

	Page
Kelly v. Robinson, 479 U.S. 36 (1986)	13
Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527 (5th Cir.), *cert. denied*, —— U.S. ——, 115 S. Ct. 82 (1994)	12
Lotus Dev. Corp. v. Paperback Software, Int'l, 740 F. Supp. 37 (D. Mass. 1990)	6
Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir.), *cert. granted*, 64 U.S.L.W. 3238 (U.S. Sept. 27, 1995) (No. 94-2003)	*passim*
M. Kramer Mfg. Co., Inc. v. Andrews, 783 F.2d 421 (4th Cir. 1986)	12
Mastro Plastics Corp. v. NLRB, 350 U.S. 270 (1956)	13
Mazer v. Stein, 347 U.S. 201 (1954)	10, 11, 17
Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)	16
Offshore Logistics, Inc. v. Tallentire, 477 U.S. 207 (1986)	13
Robert R. Jones Associates, Inc. v. Nino Homes, 858 F.2d 274 (6th Cir. 1988)	12
Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417 (1984)	8
Toro Co. v. R & R Products Co., 787 F.2d 1208 (8th Cir. 1986)	12
United States v. Heirs of Boisdoré, 49 U.S. (8 How.) 113 (1849)	13
Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987)	12
Constitutional Provision, Statutes and Public Laws	
United States Constitution, art. I, § 8, cl. 8	17
17 U.S.C. § 101	*passim*
17 U.S.C. § 102(a)	7, 9, 12
17 U.S.C. § 102(b)	*passim*
17 U.S.C. § 117	7
Pub. L. No. 98-573, 88 Stat. 1873 (1974)	8
Pub. L. No. 96-517, 94 Stat. 3015 (1980)	7, 9

TABLE OF AUTHORITIES—Continued

Legislative Materials	Page
Copyright Law Revision: Hearings on H.R. 4347, H.R. 5680, H.R. 6831 and H.R. 6835 before Subcomm. No. 3 of the House Comm. on the Judiciary, 89th Cong., 1st Sess. (1965)	8
Copyright Law Revision: Hearings Pursuant to S. Res. 37 on S. 597 before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on the Judiciary, 90th Cong., 1st Sess. (1967)	8
H.R. Rep. No. 1476, 94th Cong., 2d Sess., *reprinted in* 1976 U.S.C.C.A.N. 5659 (1976)	9, 12
House Comm. on the Judiciary, *Copyright Law Revision* (Comm. Print 1965) (1964 Revision Bill with Discussions and Comments)	8
Treaties	
Agreement on Trade-Related Aspects of Intellectual Property Rights, Including Trade in Counterfeit Goods, art. 10(1), 33 I.L.M. 81, 87 (Annex 1C to the World Trade Organization Agreement) (1994)	10
North American Free Trade Agreement, H.R. Doc. No. 159, 103d Cong., 1st Sess., ch. 17, art. 1705(1)(a) (1993), 32 I.L.M. 605, 671 (1993)	10
Other Authorities	
Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559 (1994)	14
Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977 (1993)	11
Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978)	8, 9
Information Infrastructure Task Force, *Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights* 138 (1995)	10

TABLE OF AUTHORITIES—Continued

	Page
Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* (1994)	5, 14
U.S. Dept. of Commerce, *U.S. Global Trade Outlook 1995-2000: Toward the 21st Century* (1995)	2
U.S. Dept. of Commerce, *U.S. Industrial Outlook '94* (1994)	2
Webster's Ninth New Collegiate Dictionary (1987)	14

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

AMICUS CURIAE BRIEF OF DIGITAL EQUIPMENT CORPORATION, THE GATES RUBBER COMPANY, INTEL CORPORATION AND XEROX CORPORATION IN SUPPORT OF PETITIONER

INTEREST OF AMICI CURIAE

Digital Equipment Corporation, The Gates Rubber Company, Intel Corporation and Xerox Corporation ("Amici") file this amicus curiae brief in support of petitioner Lotus Development Corporation ("Lotus").

Amici include leading developers of software and other computer products, which they create and market for mainframe, mini and personal computers for the business, education, government and entertainment markets. All of Amici develop computer software to compete in the worldwide software market, to complement the products

and services they provide their customers, or to support other segments of their businesses. Amici constitute a major segment of the U.S. economy and collectively have annual revenues exceeding $30 billion, employ more than 130,000 people worldwide, and invest more than $2 billion annually in research and development.

Amici have no interest in either of the parties to this dispute or in the outcome of this case, other than to the extent that it affects the computer software industry and the public generally. Amici have a strong interest in the isues before the Court, since all of Amici develop computer software.

This case presents a critical issue concerning the extent to which copyright protects expression in computer programs. The computer software industry has long relied on copyright law and the incentives it provides. Under the regime established by Congress that recognizes computer programs as literary works and accords them full protection under copyright law, software development has grown into one of the major American industries. At the same time, the United States has become the dominant force in the world software market, supplying approximately 75% of the nearly $70 billion worldwide packaged software market in 1993 and maintaining a positive balance of trade. U.S. Dept. of Commerce, *U.S. Global Trade Outlook 1995-2000: Toward the 21st Century* 134-35 (1995) ("Trade Outlook").

From 1991 to 1994 the market for packaged software in the United States increased by nearly 41%, to $35.6 billion, *id.*, making it one of the fastest-growing sectors in our national economy. U.S. Dept. of Commerce, *U.S. Industrial Outlook '94*, at 27-1 (1994). Employment in the software industry has risen every year since 1988 (the first year for which such data are available), creating nearly 73,000 jobs over that period. *Trade Outlook*, at 134.

SUMMARY OF ARGUMENT

This case presents the question whether Section 102(b) of the Copyright Act bars protection for expression in the user interface of a computer program. The key to resolving this question lies in the recognition and implementation of two congressional mandates found in the Copyright Act: Congress' mandate that computer programs are literary works protectible by copyright under the same principles that pertain to other literary works; and Congress' mandate that although expression in a work is protectible, the ideas, processes, methods of operation and the like that underlie that expression are not protectible.

The text and legislative history of the Copyright Act of 1976 and 1980 computer software amendments, make clear that computer programs are literary works protectible by copyright. Apart from very narrow exceptions, Congess did not establish different rules or principles to govern protection of computer programs. Rather, Congress chose to protect programs according to the same principles that apply to other literary works.

The text and legislative history also establish clearly that Section 102(b) of the Copyright Act does no more than restate the idea/expression dichotomy that has been a part of U.S. copyright law for more than a century. According to that most basic principle, expression in works is copyrightable; ideas that underlie that expression are not.

The court below has cast doubt on the protectibility of expression in computer programs. It accomplished this by defining Section 102(b)'s phrase "method of operation" in a way that overlaps substantially with Section 101's definition of "computer program," and then holding that anything falling within that definition is uncopyrightable regardless of its expressive content. This result undermines Congress' intention to provide computer pro-

gram authors with the same incentives accorded other authors under copyright.

ARGUMENT

The central issue in this case is whether Section 102 (b) of the Copyright Act of 1976, 17 U.S.C. § 102(b), renders expression of elements of computer programs uncopyrightable because those elements may be used to operate other elements of the program. The specific program element before the Court is the Lotus 1-2-3 menu command hierarchy—the selection of 469 commands and their hierarchical arrangement into more than 50 menus and submenus which collectively comprise the heart of the user interface of Lotus' 1-2-3 spreadsheet program. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 809 (1st Cir.), *cert. granted*, 64 U.S.L.W. 3238 (U.S. Sept. 27, 1995) (No. 94-2003). But this case has profound implications for all other program elements, for computer programs as a whole, and for other works as well.

A user interface is the element of a computer program that permits the program author to communicate with the user, and permits the user to communicate with the computer. This communicative aspect of a user interface makes it one of the most expressive aspects of a computer program. In the decision below, the First Circuit accepted the district court's finding that the 1-2-3 menu command hierarchy was expressive, and that other ways of expressing the ideas underlying the user interface were available. 49 F.3d at 816. Nonetheless, the court below found that the Lotus 1-2-3 menu command hierarchy—including its expression—was a "method of operation." For that reason, it held that Section 102(b) of the Copyright Act deprived it of protection.[1] 49 F.3d at 815.

[1] 17 U.S.C. § 102(b) provides:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

In the First Circuit's view, " 'method of operation,' as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer." *Id.* According to the court, anything that falls within that definition is wholly deprived of copyright protection, even to the extent that it contains original expression.[2] *Id.*, at 815-16.

Since, in the court's view, a user interface is a "means by which a person operates . . . a computer," the court's reasoning appears to preclude protection for any user interface. The First Circuit's construction of Section 102(b) would deny protection to expression in most other program elements as well, and to computer programs as a whole. A computer program, defined in the Copyright Act as "a set of statements or instructions to be *used* directly or indirectly in a computer *in order to bring about a certain result*," is a means by which a person operates a computer. 17 U.S.C. § 101 (emphasis supplied). Virtually all elements of a program are means of operating some aspect of a computer. The First Circuit's construction of Section 102(b) might well preclude protection for all such elements—even for the code, or text,

[2] We refer, of course, to expression that has not "merged" with the underlying idea. Expression is deemed to have merged with the underlying idea when there is only one way available to express the idea. See generally 3 Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* § 13.03[B][3] (1994), (hereinafter, "Nimmer"). The court below found that there were alternative ways of expressing the ideas in the Lotus menu command hierarchy (thus ruling out merger), but found that to be "immaterial" to its analysis. 49 F.3d at 816.

By referring to "original expression" we exclude *"scènes à faire"* —stock treatments of a given subject—that, if protected at all, are protected only against wholesale appropriation. See *Atari Games Corp. v. Oman*, 888 F.2d 878, 886 (D.C. Cir. 1989) (R.B. Ginsburg, J.).

of the program.[3] In his concurring opinion, Judge Boudin, in fact, acknowledged that such a literal reading of Section 102(b) "might easily seem to exclude most computer programs from protection." 49 F.3d at 820 (Boudin, J., concurring).

In determining whether a particular element of a computer program is protectible under copyright, a court is guided by two congressional mandates: (a) Congress' mandate that computer programs be protected under copyright according to the same principles as other literary works, and (b) Congress' mandate that no "idea, procedure, process, system, method of operation, concept, principle, or discovery" may be protected by copyright. These are "the markers that Congress has placed to guide the courts." *Lotus Dev. Corp v. Paperback Software, Int'l*, 740 F. Supp. 37, 53 (D. Mass. 1990). Having lost sight of these "markers," the court of appeals regarded itself as "navigating in uncharted waters," 49 F.3d at 813, and thus free to steer any course it might choose.

I. CONGRESS MANDATED THAT COPYRIGHT PROTECT THE EXPRESSION IN COMPUTER PROGRAMS, INCLUDING THE EXPRESSION OF METHODS OF OPERATION.

A. Congress Determined that Computer Programs are Literary Works Protected by Copyright Under the Same Principles that Govern Protection of Other Literary Works.

Computer programs were protected under the Copyright Act of 1976 when it was enacted, even though they were not enumerated specifically among the categories of protectible works of authorship. Section 102(a) states

[3] The court's effort to distinguish "computer code" from the Lotus menu command hierarchy, 49 F.3d at 816, rests on flawed reasoning. See *infra*, pp. 14-15.

that the Copyright Act protects "original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device." Section 102(b) then sets out a nonexclusive list of categories of works, which includes "literary works." 17 U.S.C. § 102 (a)(1). Literary works are defined as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia . . . ." 17 U.S.C. § 101.

Computer programs, whether in a form that can be read by programmers or in a form that can be executed by a computer, are "works . . . expressed in words, numbers, or other . . . symbols or indicia," and thus fall squarely within the statutory definition of literary works.

Congress later confirmed that computer programs are protected under the Coypright Act by enacting a statutory definition of "computer program" in Section 101. Pub. L. No. 96-517 §10(a), 94 Stat. 3015, 3028 (1980). It declined to create a new category of works for computer programs, leaving them in the literary works category. Congress also declined to enact special rules applicable only to computer programs, apart from the narrow exceptions permitting an owner of a copy to create a copy or adaptation "as an essential step in the utilization of the computer program in conjunction with a machine" and to make a backup copy. 17 U.S.C. § 117. It must be presumed that in all other respects Congress intended the copyright law to apply to computer programs in accordance with the same principles applicable to other literary works.

The legislative history of the Copyright Act confirms this reading of the statutory language. The Copyright Act of 1976 culminated more than three decades of consideration and debate by Congress. Congress considered many different viewpoints presented in testimony, written submissions, and studies. Protection of computer programs

figured prominently in these proceedings,[4] particularly after the Copyright Office began issuing registrations for claims of copyright in computer programs in 1964. *Final Report of the National Commission on New Technological Uses of Copyright Works* 15 (1978) ("CONTU Final Report").

Still, Congress sought further input and constituted a panel of experts, the National Commission on New Technological Uses of Copyrighted Works ("CONTU"), to study and make recommendations concerning, *inter alia*, the proper regime for protecting computer programs. Pub. L. No. 93-573, 88 Stat. 1873 (1974). Although the Copyright Act was enacted before CONTU's study was complete, the committee report prepared when Congress was deliberating on the Copyright Act [5] states plainly that

[4] See, *e.g.*, *Copyright Law Revision: Hearings on H.R. 4347, H.R. 5680, H.R. 6831 and H.R. 6835 before Subcomm. No. 3 of the House Comm. on the Judiciary*, 89th Cong., 1st Sess. 1144-50 (statement and testimony of John F. Banzhaf III), 1898-99 (letter from Graham W. McGowan to Hon. Edwin E. Willis, House Judiciary Committee), 1428, 1443, 1446-48 (testimony of Bella L. Linden) (1965); 5 House Comm. on the Judiciary, *Copyright Law Revision* 62-63 (testimony of Morton David Goldberg), 271 (statement of H.R. Mayers) (Comm. Print 1965) (1964 Revision Bill with Discussions and Comments); *Copyright Law Revision: Hearings Pursuant to S. Res. 37 on S. 597 before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on the Judiciary*, 90th Cong., 1st Sess. 192-201 (1967) (testimony and statement of Prof. Arthur R. Miller).

[5] See *Gustafson v. Alloyd Co.*, —— U.S. ——, 115 S. Ct. 1061, 1072 (1995) ("If legislative history is to be consulted, it is preferable to consult the documents prepared by Congress when deliberating."). The report of the House Judiciary Committee that accompanied the copyright revision bill has been cited frequently by courts as evidence of Congress' intent in enacting the Copyright Act. See, e.g., *Campbell v. Acuff-Rose Music, Inc.*, —— U.S. ——, 114 S. Ct. 1164, 1170 (1994); *Fogerty v. Fantasy, Inc.*, —— U.S. ——, 114 S. Ct. 1023, 1028 (1994); *Feist Publications, Inc., v. Rural Tel. Serv. Co.*, 499 U.S. 340, 355-57, 359 (1991); *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 448-49 nn.29, 31-32 (1984).

computer programs are among the literary works protected by Section 102(a)(1) of the statute:

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.

H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667 (1976) ("House Report").

CONTU took testimony and written submissions from numerous experts in copyright and information technology. *CONTU Final Report*, at 113-14, 121-29. After consideration and debate, it concluded that copyright should protect computer programs under the same principles that apply to other copyrighted works. *Id.* at 13. In 1980, Congress adopted CONTU's recommendations with immaterial changes—including CONTU's proposed definition of "computer program." [6] Pub. L. No. 96-517 § 10(a), 94 Stat. 3015, 3028 (1980), *codified* at 17 U.S.C. § 101 (definition of "computer program").

Congress' confirmation that computer programs are literary works protected under copyright law has been accepted as a model for protection of software throughout the world. In fact, it has been incorporated into international obligations accepted by a clear majority of nations, including all of the United States' major trading partners.

[6] As this brief chronology demonstrates, copyright protection for computer programs is neither an historical acc'dent nor an impulsive act that received little attention by the legislators. Copyright protection for computer programs is the result of decades of open debate and careful consideration.

For example, the North American Free Trade Agreement (NAFTA) provides that computer programs are literary works under copyright law and requires all parties (the United States, Canada and Mexico) to protect them as such. NAFTA, H.R. Doc. No. 159, 103d Cong., 1st Sess., ch. 17, art. 1705(1)(a) (1993), 32 I.L.M. 605, 671 (1993). The World Trade Organization Agreement (successor to the General Agreement on Tariffs and Trade) similarly requires protection of computer programs as literary works under copyright law. Agreement on Trade-Related Aspects of Intellectual Property Rights, Including Trade in Counterfeit Goods (TRIPS), art. 10 (1), 33 I.L.M. 81, 87 (Annex 1C to the World Trade Organization Agreement) (1994). See Information Infrastructure Task Force, *Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights* 138 (1995).

B. Section 102(b) of the Copyright Act Bars Protection Only for Ideas, Not for Expression of Ideas.

Section 102(b) of the Copyright Act is a statutory restatement of the "idea/expression" dichotomy. This principle, first enunciated by the Court in *Baker v. Selden*, 101 U.S. 99 (1879), formed part of the legal context in which the Copyright Act was considered and enacted. See, e.g., *Mazer v. Stein*, 347 U.S. 201, 217 (1954) (restating idea/expression dichotomy). The reference to "idea" in Section 102(b) is a conscious invocation of the principle.[7]

Expression that is separable from (*i.e.*, is not merged with) an idea, method of operation, or any of the other categories of unprotectible matter, is copyrightable.[8] Noth-

[7] The phrase "method of operation," as used in copyright law, also appears to originate in *Baker*. 101 U.S. at 103.

[8] For simplicity, hereafter we refer to all of the categories of unprotectible matter listed in Section 102(b) collectively as "ideas," unless the context dictates otherwise.

ing in the text of Section 102(b) states or even suggests otherwise. To the contrary, the final phrase of Section 102(b) ("regardless of the form in which it is described, explained, illustrated, or embodied in such work") contemplates that ideas, procedures, processes and the like are separable from the form in which they are "described, explained, illustrated, or embodied . . . ."

A dichotomy thus exists between unprotectible ideas and their protectible "form" or expression. Apart from those instances where the idea and expression in a particular work may have merged (another principle that owes its origin to *Baker*), the idea/expression dichotomy does not deny protection to expression in a work.[9] This was settled law at the time Congress enacted the Copyright Act, see, e.g., *Mazer*, 347 U.S. at 217 ("protection is given only to the expression of the idea—not the idea itself"); the text of Section 102(b) reveals no Congressional intent to change the law in this regard.

The legislative history of the Copyright Act confirms this reading of the statutory language:

> Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law.
>
> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged.

[9] See Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 986 (1993) ("Since *Baker*, courts have reaffirmed repeatedly that functionality poses no per se bar to copyrightability.") (The author was a member of CONTU who served on its software subcommittee.).

House Report, at 57, 1976 U.S.C.C.A.N. at 5670. See also *id.* at 54, 1976 U.S.C.C.A.N. at 5667 (computer programs are protectible under Section 102(a)(1) "to the extent that they incorporate authorship in the programmer's expression . . . as distinguished from the ideas themselves.").

Implementing Congress' express statement in the legislative history of the Copyright Act, courts of appeals have universally found Section 102(b) to be a restatement of the idea/expression dichotomy. See, e.g., *CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc.*, 44 F.3d 61, 69 & n.12 (2d Cir. 1994), *cert. denied*, —— U.S. ——, 64 U.S.L.W. 3240 (1995); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); *M. Kramer Mfg. Co., Inc. v. Andrews*, 783 F.2d 421, 434 (4th Cir. 1986); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 533 & n.8 (5th Cir.), *cert. denied*, —— U.S. ——, 115 S. Ct. 82 (1994); *Robert R. Jones Assocs., Inc. v. Nino Homes*, 858 F.2d 274, 277 (6th Cir. 1988); *Atari, Inc. v. North American Philips Consumer Elecs. Corp.*, 672 F.2d 607, 615 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982); *Toro Co. v. R & R Prods. Co.*, 787 F.2d 1208, 1211-12 (8th Cir. 1986); *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 & n.11 (9th Cir. 1994), *cert. denied*, —— U.S. ——, 115 S. Ct. 1176 (1995); *Autoskill Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 (10th Cir.), *cert. denied*, —— U.S. ——, 114 S. Ct. 307 (1993); *Cable/Home Communication Corp. v. Network Prods., Inc.*, 902 F.2d 829, 842-43 & n.26 (11th Cir. 1990).

II. IN HOLDING EXPRESSION OF A METHOD OF OPERATION TO BE UNCOPYRIGHTABLE PER SE, THE DECISION BELOW THWARTS CONGRESS' SCHEME FOR PROTECTING COMPUTER PROGRAMS.

A. The First Circuit's Per Se Bar to Protecting User Interfaces Is a Misinterpretation of the Copyright Act.

The First Circuit's holding that an element of a computer program that performs a useful function is a "method of operation," and all expression in that element is therefore ineligible for copyright protection under Section 102(b), rests on a series of misinterpretations of the Copyright Act.

The Court has long emphasized that a statutory provision should be interpreted in light of "the provisions of the whole law, and . . . its object and policy . . . ." *Kelly v. Robinson*, 479 U.S. 36, 43 (1986) (quoting *Offshore Logistics, Inc. v. Tallentire*, 477 U.S. 207, 221 (1986), in turn quoting *Mastro Plastics Corp. v. NLRB*, 350 U.S. 270, 285 (1956), in turn quoting *United States v. Heirs of Boisdoré*, 49 U.S. (8 How.) 113, 122 (1849)) (citations and internal quotes omitted). The First Circuit failed utterly to do this. Without so much as a sideways glance at its context in the statute, the First Circuit regarded Section 102(b) in isolation and proffered an interpretation of the provision that undoes what Congress did elsewhere in the statute.

While it is true that "the 'starting point in every case involving construction of a statute is the language itself,' " *Kelly*, 479 U.S. at 43 (quoting *Blue Chip Stamps v. Manor Drug Stores*, 421 U.S. 723, 756 (1975) (Powell, J., concurring)), "the text is only a starting point." *Kelly*, 479 U.S. at 43. The canon that a court "construe[s] a statutory term in accordance with its ordinary or natural meaning," *FDIC v. Meyer*, 510 U.S. ——, 114 S. Ct. 996, 1001 (1994), should not be applied to a single

phrase in isolation in a way that defeats other provisions of the statute.

The phrase "method of operation" is not defined in the Copyright Act. The court below, presumably adopting what it viewed as a common use of the words "method of operation," 49 F.3d at 821 (Boudin, J., concurring), constructed a definition for that phrase that is strikingly similar to that adopted by Congress for a "computer program" in Section 101. The court then applied Section 102(b) as a per se bar to protection for any element of a work (and presumably for any work in its entirety), regardless of its expressive content, that falls within that definition. On its face, that construction must be wrong.

First, it can be presumed that Congress did not intend to adopt for *uncopyrightable* matter a definition that has essentially the same meaning as the statutory definition of a *copyrightable* work of authorship—namely, a computer program.[10] The court of appeals has constructed a false conflict between Sections 101 and 102(b) of the Copyright Act and arbitrarily resolved it in favor of Section 102(b).

The fallacy of the First Circuit's construction is illustrated by substituting "idea" for "method of operation." "Idea" can be defined as "a formulated thought or opinion." *Webster's Ninth New Collegiate Dictionary* 597 (1987). Therefore (following the First Circuit's logic) any element of a work that fits that definition is un-

---

[10] See 1 *Nimmer*, at § 2.03[D]; Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559, 2570 (1994) ("[S]ince we know that Congress did determine in 1980 to protect computer programs, the terms 'process,' 'system,' or 'method of operation' must not be understood literally." (footnote omitted)). See also *id.* at 2570 n.64 ("some nonsoftware works of authorship are protected even though they constitute 'processes' or 'methods of operation' in the colloquial sense").

protectible, no matter how expressed. That interpretation would, of course, disqualify most writings from copyright protection.

Second, even assuming that the First Circuit's definition of "method of operation" were defensible, that would not dispose of the question whether the manner in which a "method of operation" is expressed is copyrightable. As a restatement of the idea/expression dichotomy, Section 102(b) cannot bar protection for *expression* in a work where, as here, there is no merger of idea and expression. 49 F.3d at 816.

Although the First Circuit attempted to limit its holding by distinguishing "computer code"—the text of a computer program—from the Lotus menu command hierarchy, it did so based on flawed and unpersuasive reasoning. *Id.* at 816. The court stated that the menu command hierarchy of Lotus 1-2-3 was a method of operation because "[w]ithout the menu command hierarchy, users would not be able to access and control, or indeed make use of *Lotus 1-2-3's* functional capabilities." *Id.* at 815 (emphasis supplied). On the other hand, according to the court, the code of Lotus 1-2-3 was not a method of operation because, in order for a competing program "to offer *the same capabilities as Lotus 1-2-3* [a competitor does not] have to copy Lotus' underlying code . . . ." *Id.* at 816 (emphasis supplied).

The court asked a different question regarding the menu command hierarchy (whether it was needed to operate *a particular program*—Lotus 1-2-3) than it asked regarding the program's code (whether it was needed to offer *the same capabilities* as Lotus 1-2-3). Had it asked the same question regarding the user interface that it asked regarding the code, the court of appeals would have recognized that the menu command hierarchy is eligible for copyright protection, since a program offering *the same*

capabilities as Lotus 1-2-3 could have a different menu command hierarchy, as the district court found.[11]

By the same token, had it asked the same question regarding the program's code that it asked as to the user interface, it would have recognized the flaw in its reasoning. The code of Lotus 1-2-3 is necessary to operate Lotus 1-2-3, just as the code of any program is necessary to operate that particular program, making the code—under the First Circuit's reasoning—an unprotectible method of operation.

The First Circuit's misinterpretation of the Copyright Act is further illustrated by its statement that the district court erred by limiting "Lotus 1-2-3's 'method of operation' to an abstraction." 49 F.3d at 816. That statement alludes to Judge Learned Hand's observation that

> [u]pon any work . . . a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. . . . [T]here is a point in this series of abstractions where they are no longer protected, since otherwise the [author] could prevent the use of his "ideas," to which, apart from their expression, his property is never extended.

Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931). This analytic tool, which has been applied to computer programs in a number of cases, see, e.g., *Gates Rubber Co. v. Bando Chem. Indus.*, 9 F.3d 823, 834 (10th Cir. 1993); *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 706 (2d Cir. 1992), assists a court in identifying where protectible expression ends and unprotectible idea begins. In other words, the ideas barred from protection under Section 102(b) *are* a level of abstraction within a work.

---

[11] In itself, this indicates that the Lotus menu command hierarchy is expressive. See *Atari Games*, 888 F.2d at 885 ("the variety of ways to perform the same function sustains the classification of such works as 'expression' ").

B. Denying Protection to Expression in Computer Programs Undermines the Incentives Congress Enacted for Their Authors.

The Constitution grants Congress the power "To Promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries," U.S. Const., art. I, § 8, cl. 8. It thus establishes both the "object and policy" of copyright law (to promote the progress of science—*i.e.*, knowledge) and instructs Congress as to the appropriate mechanism for achieving that goal (granting authors exclusive rights for a limited period of time as an incentive to create new works).

The Court has confirmed that the public good is served by providing authors with these exclusive rights as an incentive to create works of authorship:

> The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in "Science and useful Arts." Sacrificial days devoted to such creative activities deserve rewards commensurate with the services rendered.

Mazer, 347 U.S. at 219.

When Congress acted to protect computer programs as literary works under copyright, it implemented the constitutional dictate that the public good be served by granting exclusive rights as an incentive to authors. Experience has demonstrated the soundness of this approach for computer programs. The computer software industry relies on these incentives, and they have caused the industry to flourish and grow. Any fundamental alteration in the balance of these incentives, as in the decision below, is outside the competence of the judiciary and will have a substantial and detrimental effect on the industry.

Software development (including user interface design)[12] requires a substantial investment in creative expression—the very creativity that the economic incentives in the Copyright Act are designed to protect and nurture. Eliminating copyright protection for programs, or expressive elements of programs, undermines those incentives to innovate.[13] Even casting substantial doubt on the availability and extent of such protection stifles progress and creativity, and disserves the goals of the Copyright Act.

This is so despite the First Circuit's mistaken belief that its holding would foster progress in the software development industry. 49 F.3d at 817-18. The First Circuit was obliged to abide by the balance struck by Congress in establishing the scheme of incentives for authors. The Court's admonition to another court of appeals is appropriate in this instance as well:

> We agree with the Court of Appeals that copyright is intended to increase and not impede the harvest of knowledge. But we believe the Second Circuit gave insufficient deference to the scheme established by the Copyright Act for fostering the original works that provide the seed and substance of this harvest. The rights conferred by copyright are designed to assure contributors to the store of knowledge a fair return for their labors.

---

[12] As the most visible element of a computer program, the user interface is also the most easily plagiarized. Excluding this important programming element from protection would inevitably lead to more unauthorized copying.

[13] In *Apple Computer, Inc. v. Formula Int'l, Inc.*, 562 F. Supp. 775, 783 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984), the court observed: "F[illegible]panies are going to invest the time and resources to deve[illegible]p new programs if their products can be freely duplicated by anyo[illegible]. Such 'competitors,' who could undersell the originator simply because they don't have its development costs, would destroy the market which any innovator needs to recoup his investment."

Harper & Row, Publishers v. Nation Enters., 471 U.S. 539, 545-46 (1985) (citation omitted).

CONCLUSION

The judgment of the court below should be reversed.

Respectfully submitted,

MORTON DAVID GOLDBERG *
JUNE M. BESEK
DAVID O. CARSON
JESSE M. FEDER
SCHWAB GOLDBERG PRICE
& DANNAY
1185 Avenue of the Americas
New York, NY 10036
(212) 575-8150
Attorneys for Amici Curiae

November 9, 1995

* Counsel of Record

FILED
DEC 8 1995
CLERK

No. 94-2003

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

VS.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ Of Certiorari to the United States Court of Appeals For The First Circuit

BRIEF AMICUS CURIAE OF COPYRIGHT LAW PROFESSORS IN SUPPORT OF RESPONDENT

PAMELA SAMUELSON
(Counsel of Record)
CORNELL LAW SCHOOL
322 Myron Taylor Hall
Ithaca, NY 14853
(607) 255-2829
Attorney for Amici
Copyright Law Professors

December 1995

BOWNE OF SAN FRANCISCO, INC. • 345 SANSOME ST. • S.F., CA 94104 • (415) 362-2300

BEST AVAILABLE COPY

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	iii
I. INTERESTS OF AMICI	1
II. SUMMARY OF ARGUMENT	4
ARGUMENT	6
III. COPYRIGHT LAW EXCLUDES NOT JUST ABSTRACT IDEAS, BUT ALSO CONSTITUENT ELEMENTS OF METHODS AND SYSTEMS, FROM THE SCOPE OF PROTECTION AVAILABLE TO ORIGINAL WORKS OF AUTHORSHIP	6
A. Section 102(b), Which Excludes Methods Of Operation As Well As Abstract Ideas From the Scope of Copyright Protection, Was Intended To Preclude Copyright Protection For Methods Embodied In Computer Programs	7
B. Under *Baker v. Selden*, Copyright Protection Is Unavailable To Constituent Elements of Methods and Systems Embodied In a Protected Work	8
C. Many Other Recent Computer Program Cases Have Applied Section 102(b) and Precedents Such As *Baker v. Selden* To Exclude Constituent Elements of Methods and Processes From the Scope of Copyright	11
D. The Court of Appeals Was Correct In Concluding That The "Patterns of Abstractions" Approach Is Not The Only Method Of Judging Infringement In Computer Program Cases	12

TABLE OF CONTENTS (Continued)

Page

IV. METHODS AND PROCESSES EMBODIED IN PROGRAMS SHOULD NOT BE PROTECTED BY COPYRIGHT BECAUSE THEIR UTILITARIAN CHARACTER MAKES THEM MORE APPROPRIATE FOR REGULATION BY THE PATENT SYSTEM 16

A. *Baker v. Selden* Requires That Useful Methods and Systems Embodied in Copyrighted Works Should Be Protected, If At All, Only If They Meet The Standards For Patentability 16

B. Authorities Concur With *Baker* That Courts Must Defer To Patent Law To Protect Useful Methods and Systems Depicted or Embodied in Copyrighted Works 17

C. Computer Programs Embody Many Useful Methods and Systems That Should Be Protected, If At All, By The Patent System 21

D. The District Court Improperly Applied Copyright To Protect Useful Methods or Systems In The Lotus Program 22

V. CONCLUSION 25

TABLE OF AUTHORITIES

CASES

Page(s)

Apple Computer, Inc. v. Microsoft, Inc.,
35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S.Ct. 1176 (1995) 4

Apple Computer, Inc. v. Microsoft, Inc.,
799 F. Supp. 1006 (N.D. Cal. 1992) *aff'd.*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S.Ct. 1176 (1995) 11

Ashton-Tate Corp. v. Ross,
916 F.2d 516 (9th Cir. 1990) 3, 12

Atari Games Corp. v. Nintendo of America, Inc.,
30 U.S.P.Q.2d (BNA) 1401 (N.D. Cal 1993) .. 21

Atari Games Corp. v. Nintendo of America, Inc.,
975 F.2d 832 (Fed. Cir. 1992) 22

Atari, Inc. v. North American Philips Consumer Elec. Corp.,
672 F.2d 607 (7th Cir. 1982) 25

Baker v. Selden, 101 U.S. 99 (1879) ..... *passim*

Bonito Boats, Inc. v. Thunder Craft Boats, Inc.,
489 U.S. 141 (1989) 17, 20

Brief English Systems, Inc. v. Owens,
48 F.2d 555 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931) 3, 10, 23

Combustion Eng'g, Inc. v. Murray Tube Works, Inc.,
222 U.S.P.Q. (BNA) 239 (E.D. Tenn. 1984) ... 18

Computer Associates Int'l, Inc. v. Altai, Inc.,
982 F.2d 693 (2d Cir. 1992) .....*passim*

Computer Associates Int'l, Inc. v. Altai, Inc.,
775 F. Supp. 544 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992) 3-4, 12

Continental Casualty Co. v. Beardsley,
253 F.2d 702 (2d Cir.), *cert. denied*, 358 U.S. 816 (1958) 8

TABLE OF AUTHORITIES (CONTINUED)

CASES

Page(s)

Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340 (1991) 10, 20, 26

Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) 12, 15

Griggs v. Perrin, 49 F. 15 (C.C.N.D.N.Y. 1892) ... 23

In re Abele, 684 F.2d 902 (C.C.P.A. 1982) 21

In re Lowry, 32 F.3d 1579 (Fed. Cir. 1994) 21

In re Meyer, 688 F.2d 789 (C.C.P.A. 1982) 21

Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. (BNA) 124 (E.D. Mich. 1979) ... 10, 20

Kern River Gas Transmission Co. v. Coastal Corp., 899 F.2d 1458 (5th Cir.), *cert. denied*, 498 U.S. 952 (1990) 10

Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) 3, 6, 15, 22

Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 80 (D. Mass. 1990) 6, 9, 13, 16

Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992) 13, 14, 23

Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993) 24

Lotus Dev. Corp. v. Paperback Int'l, Inc., 740 F. Supp. 37 (D. Mass. 1990) 3, 9, 13, 23

Mazer v. Stein, 347 U.S. 201 (1954) 16, 17

Muller v. Triborough Bridge Auth., 43 F. Supp. 298 (S.D.N.Y. 1942) 18

NEC v. Intel Corp., 10 U.S.P.Q.2d (BNA) 1177 (N.D. Cal. 1989) .. 8

National Cloak & Suit Co. v. Standard Mail Order Co., 191 F. 528 (S.D.N.Y. 1911) 18

TABLE OF AUTHORITIES (CONTINUED)

CASES

Page(s)

Nichols v. Universal Pictures, 45 F.2d 119 (1931) 12-13, 14, 15

Sega Enterprises, Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) 11, 12, 15, 20, 22

Signo Trading Int'l v. Gordon, 535 F. Supp. 362 (N.D. Cal. 1981) 8

Taylor Instrument v. Fawley Brost Co., 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944) 11, 17

STATUTES

17 U.S.C. § 101 ..... 8, 24

17 U.S.C. § 102(a) 4, 6

17 U.S.C. § 102(b) ..... *passim*

17 U.S.C. § 113(b) 18

17 U.S.C. § 117 8

35 U.S.C. § 101 21

35 U.S.C. § 171 17

H.R. Rep. No. 1476, 94th Cong., 2d Sess. (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659 8

S. Rep. No. 473, 94th Cong., 2d Sess. (1976) 8

MISCELLANEOUS

Horace Ball, *The Law of Copyright and Literary Property* 125-28 (1944) 18-19

Ralph S. Brown, *Eligibility for Copyright Protection: A Search for Principled Standards*, 70 Minn. L. Rev. 579 (1985) 19

Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 So. Cal. L. Rev. 1091 (1995) 20

TABLE OF AUTHORITIES

MISCELLANEOUS (CONTINUED)

	Page(s)
Final Report of the National Commission on New Technological Uses of Copyrighted Works 18-23 (1978)	8, 24
Paul Goldstein, *Copyright Principles, Law & Practice* § 8.5 (1989)	19
Hearings on S. 597 Before the Senate Subcomm. On Patents, Trademarks, & Copyrights of the Comm. On the Judiciary, 90th Cong., 1st Sess. 197 (1967)	7
Benjamin Kaplan, *An Unhurried View of Copyright* (1967)	9, 18
Mark A. Lemley, *Convergence in the Law of Software Copyright?*, 10 High Tech. L.J. 1 (1995)	20
Melville B. Nimmer & David Nimmer, 1 *Nimmer On Copyright* § 2.18[A], [B] (1995)	19
J. H. Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research*, 42 Vand. L. Rev. 639 (1989)	9, 19
Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback*, 55 Law & Contemp. Probs. 311 (1992)	19, 23
Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308 (1994)	24, 25
Arthur Weil, *American Copyright Law* (1917)	19

No. 94-2003

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

vs.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF AMICUS CURIAE OF COPYRIGHT LAW PROFESSORS IN SUPPORT OF RESPONDENT

I. INTERESTS OF AMICI

This brief amicus curiae is submitted by thirty-four professors who teach and write about copyright law at accredited law schools in the United States.[1] None of the

[1] They include: Professors Keith Aoki, University of Oregon School of Law, Stephen R. Barnett, University of California at Berkeley School of Law, Ralph S. Brown, Yale Law School, Dan L. Burk, Seton Hall University School of Law, Stephen L. Carter, Yale Law School, Margaret Chon, Syracuse University College of Law, Amy B. Cohen, Western New England College School of Law, Julie E. Cohen, University of Pittsburgh School of Law, Paul J. Heald, University of Georgia

signatories to this amicus brief has an interest in either party or in the outcome of this case except to the extent that this case will have an important impact on the development of copyright law and principles and on the delicate balance between innovation and competition that intellectual property law seeks to maintain. The consent of the attorneys for both parties to file this brief has been obtained. The signatories to this brief exercised complete control over its editorial contents; Borland helped to defray the costs of printing this brief.

This brief sets forth a number of principles that we, as professors who teach and write about copyright law, believe should be considered in determining the proper scope of copyright protection for computer programs and their user interfaces. Amici do not hold identical views on all issues arising from the application of copyright law to computer programs. There are some among us who would urge the

---

School of Law, Peter A. Jaszi, American University, Washington College of Law, Beryl Jones, Brooklyn Law School, John A. Kidwell, University of Wisconsin Law School, Edmund W. Kitch, University of Virginia Law School, Robert Kreiss, University of Dayton School of Law, Roberta R. Kwall, DePaul University College of Law, Leslie Kurtz, University of California at Davis School of Law, David L. Lange, Duke University School of Law, Mark A. Lemley, University of Texas School of Law, Marshall Leaffer, University of Toledo College of Law, Jessica Litman, Wayne State University Law School, Charles R. McManis, Washington University School of Law, Neil Netanel, University of Texas School of Law, L. Ray Patterson, University of Georgia School of Law, Henry H. Perritt, Jr., Villanova Law School, David G. Post, Georgetown University Law Center, Jerome H. Reichman, Vanderbilt Law School, David A. Rice, Rutgers, The State University of New Jersey, S.I. Newhouse Center for Law & Justice, Pamela Samuelson, Cornell Law School, David J. Seipp, Boston University School of Law, David E. Shipley, University of Kentucky College of Law, Lionel S. Sobel, Loyola Law School, Lloyd Weinreb, Harvard Law School, Alfred C. Yen, Boston College Law School, and Diane L. Zimmerman, New York University School of Law.

Court to resolve the dispute on one or a few of the following grounds:

(1) a menu command hierarchy is too abstract an element of a computer program to be protectable by copyright law under 17 U.S.C. § 102(b), see *Ashton-Tate Corp. v. Ross*, 916 F.2d 516, 521-22 (9th Cir. 1990);

(2) a menu command hierarchy is unprotectable under § 102(b) as a constituent element of a method of operating a computer to perform spreadsheet functions, see *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), Pet.App 14a-16a (*Borland*), *cert. granted*, 116 S.Ct. 39 (1995);

(3) the menu command hierarchy of Lotus 1-2-3 is unprotectable under § 102(b) as "a fundamental part of the functionality" of the Lotus macro system, see *Lotus Dev. Corp. v. Paperback Software Int'l, Inc.*, 740 F. Supp. 37, 65 (D. Mass. 1990), Pet.App. 229a (*Paperback*);

(4) the menu command hierarchy of Lotus 1-2-3 is an inseparable part of a language for constructing macros, and languages are uncopyrightable under § 102(b), see, for example, *Brief English Systems, Inc. v. Owens*, 48 F.2d 555, 556 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931);

(5) the menu command hierarchy of Lotus 1-2-3 is unprotectable by copyright law under § 102(b) because it operates as a program-to-program interface in relation to the execution of macros, and is, therefore, among the elements of the Lotus program whose design was constrained by external factors, see *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 709-710 (2d Cir. 1992);

(6) the menu command hierarchy of Lotus 1-2-3 is unprotectable by copyright law under § 102(b) because it is an inseparable part of the behavior of the Lotus program which is an unprotectable process, see *Computer Associates*

Int'l. Inc. v. Altai, Inc., 775 F. Supp. 544, 560 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693, 706 (2d Cir. 1992); or

(7) when user interfaces of computer programs lie closer to the functional than to the aesthetic end of the spectrum of potential expressiveness, as the Lotus 1-2-3 user interface does, courts should employ a virtual identity standard in judging copyright infringement, *see Apple Computer, Inc. v. Microsoft, Inc.*, 35 F.3d 1435, 1444 (9th Cir. 1994), *cert. denied*, 115 S.Ct. 1176 (1995).

Even so, we agree that the traditional principles of copyright law set forth in the remainder of this brief should be employed in judging the scope of copyright protection available to the Lotus 1-2-3 menu command hierarchy in the context of this case.

II. SUMMARY OF ARGUMENT

There are two Congressional mandates courts must heed in judging copyright infringement in cases involving computer programs: (1) to protect original expression in computer programs under 17 U.S.C. § 102(a), and (2) not to protect abstract ideas or constituent elements of procedures, processes, systems and methods of operation embodied in programs, *id.* at § 102(b).

Congress had computer programs in mind when it enacted § 102(b) in 1976. Congress added this provision out of concern that without it, some software developers might attempt to claim copyright protection for commercially valuable methods or processes embodied in programs when all Congress intended to protect was original expression in program texts.

Section 102(b) codifies the principal holding of *Baker v. Selden*, 101 U.S. 99 (1879), that the constituent elements of a useful art, such as a bookkeeping system or a procedure for treating a disease, are unprotectable by copyright law, even

when embodied in an original work of authorship that qualifies for copyright protection. In keeping with the Supreme Court's holding in *Baker*, § 102(b) excludes from the scope of copyright protection not only abstract elements of works, such as ideas, concepts, principles and discoveries, but also constituent elements of "procedure[s], process[es], system[s], [and] method[s] of operation . . . embodied in [a] work." *Id.*

An important reason for excluding the constituent elements of processes and methods from the scope of copyright law is, as *Baker* noted, to avoid disrupting the competition policy principles of patent law which would occur if a court allowed an author to use copyright law to protect elements of his or her work that should be protected, if at all, by patent law. Unlike conventional literary works, computer programs contain many components that are potentially patentable. It is, thus, appropriate for courts to filter out not only abstractions, but also constituent elements of useful processes and methods of operation before making a comparison of two computer programs for the purpose of determining whether infringement has occurred.

If computer programs need more protection against competitive imitation than copyright, supplemented by patent and trade secrecy law, can provide, software developers should seek additional legal protection for computer programs from Congress. Even though incentive-based arguments for extending copyright law to reach all commercially valuable aspects of computer programs may have some appeal, adopting such a rule would ultimately have "a corrosive effect on certain fundamental tenets of copyright doctrine." *Altai*, 982 F.2d at 712.

ARGUMENT

III. COPYRIGHT LAW EXCLUDES NOT JUST ABSTRACT IDEAS, BUT ALSO CONSTITUENT ELEMENTS OF METHODS AND SYSTEMS, FROM THE SCOPE OF PROTECTION AVAILABLE TO ORIGINAL WORKS OF AUTHORSHIP.

There are two Congressional mandates that courts must heed when judging infringement in computer program cases. Courts must protect original expression in programs under 17 U.S.C. § 102(a), but must not extend protection to abstract ideas or to procedures, processes, systems, and methods of operation in programs, *id.* at § 102(b).

Neither the District Court nor the Court of Appeals in this case has satisfactorily reconciled these two mandates. The District Court viewed §102(b) too narrowly by regarding it as excluding from the scope of copyright only abstract elements of programs. *See, e.g., Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 80, 91 (D. Mass. 1990), Pet.App. 167a (*Borland I*) (" '[p]rocess,' like 'idea,' is an abstraction. . ."). The Court of Appeals correctly rejected this interpretation of § 102(b) on the ground that constituent elements of methods of operation are unprotectable under § 102(b). *Borland*, 49 F.3d at 816 (1st Cir.), Pet.App. 16a-18a. However, the Court of Appeals did not provide much guidance about how to distinguish protectable expression from unprotectable methods of operation. This brief will offer some guidance about how to reconcile the two Congressional mandates.

A. Section 102(b), Which Excludes Methods Of Operation As Well As Abstract Ideas From the Scope of Copyright Protection, Was Intended To Preclude Copyright Protection For Methods Embodied In Computer Programs.

Section 102(b) of Title 17 of the U.S. Code provides: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."

The legislative history of this provision indicates that Congress had computer programs in mind when adopting it and meant for it to limit the scope of copyright protection available for computer programs. During legislative hearings leading up to enactment of the Copyright Act of 1976, concerns had been expressed about the need for such a provision so that copyright would not overprotect computer programs.[2] Both the House and Senate Committee reports plainly state:

> Some concern has been expressed lest copyright in computer programs should extend protection to the *methodology or processes* adopted by the programmer, rather than merely to the *'writing'* expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the *actual processes or methods*

---

[2] Hearings on S. 597 Before the Senate Subcomm. on Patents, Trademarks, & Copyrights of the Comm. on the Judiciary, 90th Cong., 1st Sess. 197 (1967).

embodied in the program are not within the scope of copyright law.[3]

The legislative history of section 102(b) also indicates that the provision was intended to codify a long line of copyright cases, beginning with *Baker v. Selden*, 101 U.S. 99, that had held that constituent elements of systems, processes, and the like were beyond the scope of copyright protection available to an original work of authorship.[4] The Final Report of the National Commission on New Technological Uses of Copyrighted Works (CONTU) cited § 102(b) and functional writing cases such as *Baker* and *Continental Casualty Co. v. Beardsley*, 253 F.2d 702 (2d Cir.), *cert. denied*, 358 U.S. 816 (1958) as among the sources of guidance for courts judging copyright infringement claims involving computer programs.[5]

B. Under *Baker v. Selden*, Copyright Protection Is Unavailable To Constituent Elements of Methods and Systems Embodied In a Protected Work.

Courts sometimes describe the Supreme Court's decision in *Baker v. Selden*, 101 U.S. 99, as a case about the

[3]H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted* in 1976 U.S.C.C.A.N. 5659, 5670 and S. Rep. No. 473, 94th Cong., 2d Sess. 54 (1976) (emphasis added).

[4]*Id.* (indicating an intent to restate the caselaw). Among the cases regarding section 102(b) as codifying the principal holding of *Baker v. Selden* are *NEC v. Intel Corp.*, 10 U.S.P.Q. 2d (BNA) 1177, 1179 (N.D. Cal. 1989) and *Signo Trading Int'l v. Gordon*, 535 F. Supp. 362, 365 (N.D. Cal. 1981).

[5]Final Report of the National Commission on New Technological Uses of Copyrighted Works 18-23 (1978). Congress relied on the CONTU Report in enacting the amendments to copyright law that defined "computer program" (17 U.S.C. § 101) and provided for some special rules for programs (17 U.S.C. § 117), thereby giving the report a measure of Congressional imprimatur. *See, e.g., Altai*, 982 F.2d at 703 (deferring to CONTU Report).

idea/expression distinction. In *Paperback*, for example, the District Court described *Baker v. Selden* as having held that "the text of a book describing a special method of double-entry accounting on paper spreadsheets . . . was copyrightable *expression*, but that the . . . *idea* of this particular kind of double-entry bookkeeping, was not." *Paperback*, 740 F.Supp. at 55, Pet.App. 207a-208a; *Borland I*, 788 F.Supp. at 90-93, Pet.App. 164a-172a.

This characterization of Baker misses the forest for the trees. *Baker v. Selden* is not principally a case about the unprotectability of abstract ideas,[6] as is evident from the Court's very different statement of the basis of its decision. The Court held that Selden's copyright protected his "explanation" of the useful art described in the work, but not the "useful art" itself. *Baker*, 101 U.S. at 105. The bookkeeping system, as reflected in the particular selection and arrangement of columns and headings constituting the sample ledger sheets in Selden's book, was beyond the scope of copyright. "[T]he mere copyright of Selden's book did not confer upon him the exclusive right to make and use account-books, ruled and arranged as designated by him and described and illustrated in said book." *Id.* at 107.[7]

[6]The idea/expression distinction as related to abstractions predates *Baker v. Selden*. *See, e.g.*, J. H. Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research*, 42 Vand. L. Rev. 639, 693 n.288 (1989). Those who look to *Baker v. Selden* as the case first enunciating the idea/expression distinction are misinformed about copyright history.

[7]Although Baker's competing ledger sheets were somewhat different than Selden's in their selection and arrangement of columns and headings, the above-quoted statement makes clear that the Court would have ruled no differently had Baker used identical ledger sheets. *See also* Benjamin Kaplan, *An Unhurried View of Copyright* 64 (1967) ("the [*Baker*] privilege extends to exact copies").

The Court explained that Selden's copyright no more gave him an exclusive right to the bookkeeping system than the copyright on a book about the composition and use of medicines would give its author an exclusive right to the medicines discussed in the book. *Id.* at 102. Selden's claim had seemed plausible because the useful art in that case was embodied in a writing, whereas most useful arts are embodied in wood, metal, or stone. *Id.* at 105. However, "the principle is the same in all [cases]," said the Court. "The description of the [useful] art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself." *Id.*

This statement of the rule in *Baker* makes clear what is sometimes obscured by shorthand characterizations of *Baker* as concerned with the idea/expression distinction: *Baker* is fundamentally a case about the unprotectability of the functional content embodied in copyrighted works and the right of others to copy that content in order to make use of it.[8] In reliance on *Baker*, a long line of copyright cases have held that constituent elements of methods, systems and processes are beyond the scope of copyright protection. *See, e.g., Brief English*, 48 F.2d 555 (shorthand system not protectable by copyright); *Kepner-Tregoe, Inc. v. Carabio*, 203 USPQ (BNA) 124 (E.D. Mich. 1979) (system for teaching problem-solving techniques not protectable by copyright law); *Kern River Gas Transmission Co. v. Coastal Corp.*, 899 F.2d 1458 (5th Cir.), *cert. denied*, 498 U.S. 952 (1990) (design for gas pipeline held unprotectable by copyright law); *Taylor Instrument Cos. v. Fawley Brost Co.*, 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944) (temperature recording system held unprotectable by copyright law). It is this aspect of *Baker v. Selden* that is now codified in 17 U.S.C. § 102(b).

[8] As this Court observed in *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 350 (1991), the ability of second comers to appropriate unprotectable material from copyrighted works is not an unfair or unfortunate consequence of copyright law, but rather "the means by which copyright advances the progress of science and art."

To speak of *Baker v. Selden* as a case that is only concerned with the unprotectability of abstract ideas is to miss the main point of the case.

C. Many Other Recent Computer Program Cases Have Applied Section 102(b) and Precedents Such As *Baker v. Selden* To Exclude Constituent Elements of Methods and Processes From the Scope of Copyright.

In *Altai*, 982 F.2d at 704, the Second Circuit Court of Appeals described computer programs as having an "essentially utilitarian nature" which must be taken into account when judging infringement in computer program cases. It looked to *Baker v. Selden* as the "doctrinal starting point" in cases involving copyright in process-oriented texts, such as computer programs. *Id.*; *see also Sega Enterprises, Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) (describing computer programs as in essence, utilitarian articles and looking to *Baker* as a key precedent for judging infringement as to works having "strong functional elements"). In line with *Baker v. Selden* and its progeny, many cases involving computer programs have excluded from the scope of copyright protection available to these works constituent elements of procedures, processes, systems, and methods of operation embodied in them.

Sometimes the detailed program components held to be unprotectable under § 102(b) have been parts of a user interface. *See, e.g., Apple Computer, Inc. v. Microsoft, Inc.*, 799 F. Supp. 1006, 1038 (N.D. Cal. 1992), *aff'd*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S.Ct. 1176 (1995)

(particularized methods for presenting different views of objects and for presenting dialog boxes when selecting menu items); *Ashton-Tate*, 916 F.2d at 521-22 (spreadsheet command structure). Sometimes the methods and systems held unprotectable under § 102(b) have been embedded in the text of the program. *See, e.g., Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 845 (10th Cir. 1993) (algorithm embodied in program); *Sega*, 977 F.2d at 1522 (particularized requirements for achieving compatibility). Still other times, the process that courts have regarded as unprotectable under § 102(b) has been a program's functional behavior, that is, the results generated when program instructions are executed. *See, e.g., Altai*, 775 F. Supp. at 560 and *Gates Rubber*, 9 F.3d at 836.

While the boundary between protectable expression and unprotectable method has not always shimmered with clarity in these cases, none of these courts shied away from identifying methods and processes as unprotectable elements of copyrighted programs under § 102(b) on the ground that if they regarded this or that element of a program as an unprotectable method or process, such a ruling would substantially undermine the availability of copyright protection for computer programs.

D. The Court of Appeals Was Correct In Concluding That The "Patterns of Abstractions" Approach Is Not The Only Method Of Judging Infringement In Computer Program Cases.

Although courts have been making progress in distinguishing protectable from unprotectable elements in computer programs on a case-by-case basis, the task can be very difficult, as can be seen from the prodigious effort undertaken by the District Court in this case to reconcile *Baker v. Selden* with the "patterns of abstractions" approach to determining infringement in copyright cases derived from Judge Learned Hand's decision in *Nichols v. Universal Pictures*, 45 F.2d 119,

121 (2d Cir. 1930), *cert. denied* 282 U.S. 902 (1931). The District Court ultimately decided that *Baker v. Selden* had been supplanted by the *Nichols* "abstractions" approach. *Borland I*, 788 F. Supp. at 92-93, Pet.App. 170a. This led the District Court to construct a test for judging the copyrightability of the Lotus command hierarchy focused on the *Nichols* abstractions approach.[9] *Id.* at 90.

---

[9]To illustrate differences between the District Court's analysis in the *Borland* opinions and the Supreme Court's analysis in *Baker v. Selden*, we think it instructive to apply the District Court's "copyrightability" test to the facts of *Baker*.

The first step of the District Court's method of analysis involves construction of a hierarchy of abstractions for the aspect of the work whose "copyrightability" is to be determined. *Borland I*, 788 F. Supp. at 90, Pet.App. 164a (test formulated); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 216-19 (D. Mass. 1992), Pet.App. 128a-131a (*Borland II*) (test applied). If one constructed a hierarchy of abstractions for the ledger sheet portion of Selden's accounting book, the selection of certain words as headings and the arrangement of them and the columns under which entries would be made would seem to be at the lowest level of abstraction. Under the District Court's methodology, this would suggest that these aspects of the ledger sheets should be regarded as expressive.

Consistent with the second step of the District Court's methodology, one would not inquire whether this selection and arrangement was a constituent element of a system, but rather whether the idea of an accounting system and Selden's expression of that idea were merged. To discern this, a court following the District Court's lead would likely examine other accounting books available in the marketplace, just as the District Court considered the availability of other designs for spreadsheet programs in *Paperback* and *Borland*. *See, e.g., Paperback*, 740 F. Supp. at 65-69, Pet.App. 229a-235a; Borland II, 799 F. Supp. at 217-18, Pet.App. 131a-132a. A study of these other accounting materials would likely reveal that Selden's particular selection and arrangement of elements were not essential to every expression of the idea of an accounting system, or solely dictated by the accounting functions they were to perform. Insofar as Selden had freedom of choice

Application of this test caused the District Court to conclude that Borland had infringed Lotus' copyright because the Lotus command hierarchy was among the more particularized elements of the Lotus program, and many choices existed as to the selection and arrangement of command terms, so there was no merger of idea and expression. *Borland II*, 799 F. Supp. at 216-19, Pet.App. 131a-135a.

With all due respect to the District Court's herculean effort to formulate an abstractions-based approach to judging infringement in computer program copyright cases, we think the District Court overgeneralized from the *Nichols* decision and failed to appreciate the continuing viability of *Baker v. Selden*. As we have shown, *Baker* mandates exclusion of particularized functional content, such as systems and methods of operation, from the scope of copyright

---

about which words to use and in what order to arrange them, under the District Court's methodology, they would likely be treated as expressive rather than being merged with the idea of an accounting system.

In line with the third step of the District Court's methodology, one would next inquire whether the ledger sheets were a substantial or nontrivial component of Selden's book. The answer to this question would almost certainly be that they were a substantial part. Because Baker's ledger sheets were substantially similar to Selden's, infringement would probably have been found applying the District Court's methodology to the facts of *Baker v. Selden*.

We see nothing in the Supreme Court's opinion in *Baker v. Selden* to suggest that the Court's ruling depended on whether Selden invented his system *before* selecting and arranging the columns and headings in the ledger sheets, developed the system and the ledger sheets simultaneously, or even developed the ledger sheets first and then decided later that they would permit people to do better accounting in accordance with a system they suggested. The District Court seems to suggest that merger can only be found if the system existed first and dictated the arrangement of elements. See *Borland II*, 799 F. Supp. at 212-13, Pet.App. 121a-123a. Nothing in *Baker* supports this distinction.

protection in cases involving functional writings such as computer programs. If Judge Hand did not see fit to mention the exclusion of systems, processes, and the like from the scope of copyright protection in *Nichols*, it was likely because the work at issue—a dramatic play—was of an artistic and fanciful character. Such works are predominantly expressive in content, and so enjoy a broad scope of copyright protection. Only their more abstract elements must be filtered out before an infringement analysis is done. Since functional writings, by definition, contain not only abstract ideas, but also uncopyrightable elements, such as procedures, processes, systems, and methods of operation, the scope of copyright protection available to such works tends to be narrower than for artistic and fanciful works. *See, e.g., Sega*, 977 F.2d at 1524. The functional methods or systems in these writings must be filtered out before assessing substantial similarity for infringement purposes, as many computer cases recognize. *See, e.g., Altai*, 982 F.2d at 707-711; *Gates Rubber*, 9 F.3d at 845.

The First Circuit Court was correct in asserting that it can be misleading to employ an abstractions approach in a case such as this one where the principal issue is whether the aspect of the program that has been copied is or is not an integral part of an unprotectable method or system. *Borland*, 49 F.3d at 815, Pet.App. 14a-15a. It was also correct in holding, in reliance on *Baker v. Selden*, that " 'methods of operation' are not limited to mere abstractions." *Id.* at 815-16, Pet.App. 17a. The First Circuit may not have provided an optimal degree of guidance in distinguishing between the unprotectable method it saw in the Lotus program and the protectable expression in the Lotus program, but its decision is consistent with the line of cases described above in which courts have engaged in case-by-case assessments about elements of computer programs that should be regarded as unprotectable methods or processes under *Baker* and § 102(b).

IV. METHODS AND PROCESSES EMBODIED IN PROGRAMS SHOULD NOT BE PROTECTED BY COPYRIGHT BECAUSE THEIR UTILITARIAN CHARACTER MAKES THEM MORE APPROPRIATE FOR REGULATION BY THE PATENT SYSTEM.

A. *Baker v. Selden* Requires That Useful Methods and Systems Embodied in Copyrighted Works Should Be Protected, If At All, Only If They Meet The Standards For Patentability.

An important factor in the Supreme Court's decision in *Baker v. Selden* was Selden's apparently unsuccessful effort to secure a patent on his bookkeeping system. *See Baker*, 101 U.S. at 104 (Selden had no patent); 25 L.Ed. 841, 841 (Selden tried to get patent). Baker's lawyer used Selden's quest for a patent to argue that Selden's system was "a contribution to [the] useful, mechanical art[s], not to literature." *See id.*

The Court agreed that Selden was trying to get indirectly—through a copyright infringement action—a kind of protection that he had been unable to get directly from the Patent Office, namely, an exclusive right in the system. *Baker*, 101 U.S. at 104-05. As the Court explained, "[t]o give to the author of the book an exclusive property in the [useful] art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters patent, not of copyright." *Id.* at 102.[10] This Court has

[10] In *Borland I*, 788 F. Supp. at 91, Pet.App. 167a, the District Court stated that "[t]he mere fact that patent law allows a means of legal protection for a process . . . does not establish that there is not also some protection in copyright law." It cited the Supreme Court decision in *Mazer v. Stein*, 347 U.S. 201 (1954) as a precedent recognizing an overlap of copyright and patent protection. *Mazer*, however, involved a subject matter that was potentially eligible for both copyright and design patent protection. The statuette in that case was eligible for copyright protection as a sculpture. Because it was intended for sale as a lamp base, it was also potentially eligible for protection as a new and original ornamental design for an article of manufacture under design patent law, 35 U.S.C. § 171. We do not read *Mazer* as recognizing an overlap of copyright and utility patent subject matter, and we know of no precedent which recognizes the coexistence of copyright and utility patent in the same aspect of the same work. To the contrary, several courts have relied on *Baker* to reject claims of copyright infringement for aspects of a work that were the subject of expired utility patents. *See, e.g., Taylor*, 139 F.2d 98. Both copyright and patent protection can, of course, be available to computer programs, but each law protects different aspects of programs.

consistently held that innovations in the useful arts that do not meet patent law's novelty and nonobviousness standards, are, if revealed in a publicly circulated product, freely available to be copied by competitors. *See, e.g., Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141 (1989).

B. Authorities Concur With *Baker* That Courts Must Defer To Patent Law To Protect Useful Methods and Systems Depicted or Embodied in Copyrighted Works.

The principle of deferring to patent law for protection of useful methods and the like has been applied in a long line of copyright cases sanctioning the right of competitors to reproduce functional content regardless of the tangible medium in which it was first fixed. Competitors have, for example, been allowed to copy such things as the detailed design of a three-dimensional boiler or an article of clothing, even though this might require making a schematic repre-

sentation similar to the plaintiff's design document.[11] This line of cases, now codified in 17 U.S.C. § 113(b), reflects the general principle laid down in *Baker* that the exclusive reproduction right of copyright cannot prevent—directly or indirectly—the use of unprotected ideas or utilitarian features embodied in protected works.

Baker makes clear that this principle applies to functional writings regardless of whether they are expressed in a literary or graphic form:

> The fact that the art described in the book by illustrations of lines and figures which are reproduced in practice in the application of the art, makes no difference. . . . Had he used words of description instead of diagrams. . .there could not be the slightest doubt that others, applying the art to practical use, might lawfully draw the lines and diagrams. . .which he [the author] thus described by words in his book.
>
> The copyright of a work on mathematical science cannot give. . .an exclusive right to the methods of operation. . .or to the diagrams which he employs to explain them, so as to prevent an engineer from using them whenever occasion requires. *Baker*, 101 U.S. at 103.

Later commentators, including Professor Benjamin Kaplan of Harvard, have understood this statement to recognize a privilege to appropriate uncopyrightable content from scientific and functional writings.[12]

---

[11] *See, e.g., Combustion Eng'g, Inc. v. Murray Tube Works, Inc.*, 222 U.S.P.Q. (BNA) 239, 244 (E.D. Tenn. 1984); *Muller v. Triborough Bridge Auth.*, 43 F. Supp. 298 (S.D.N.Y. 1942); *National Cloak & Suit Co. v. Standard Mail Order Co.*, 191 F. 528 (C.C.S.D.N.Y. 1911).

[12] *See, e.g.*, Benjamin Kaplan, *An Unhurried View of Copyright* 63-66 (1967); Horace Ball, *The Law of Copyright and Literary Property* 125-28, 274-78 (1944); Arthur Weil, *American Copyright Law* 191, 209, 411-12 (1917). *See also* Ralph S. Brown, *Eligibility for Copyright Protection: A Search for Principled Standards*, 70 Minn. L. Rev. 579, 600-06 (1985); Reichman, *Programs As Know-How, supra* note 6, at 693 n. 288; Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback*, 55 Law & Contemp. Probs. 311, 326-27, n.70-71 (1992). *But see* Melville B. Nimmer & David Nimmer, 1 *Nimmer On Copyright* § 2.18[A], [B] (1995).

The very fact that functional innovations are so valuable explains why copyright law, with its low standards for obtaining protection and its long duration of exclusive rights, should not protect them. As Professor Goldstein has observed:

> Functional works [such as architectural plans, legal forms, and computer programs] depend for their value primarily on the ingenuity, accuracy, and efficiency—the utility—of their underlying system, concept or method. As a result, enforcement of copyright in these works inevitably threatens the fundamental precept that copyright protection shall not extend to any "idea, procedure, process, system, method of operation, concept, principle, or discovery." Copyright in functional works is in this respect like copyright in fact works, which pose the similar danger of monopolizing elements that should be available for free use by all. The important difference is that in protecting works that are essentially functional in nature, copyright may contradict the principle that protection for these utilitarian elements is better left to the more exacting standards of patent and trade secret law.

Paul Goldstein, *Copyright Principles, Law & Practice* § 8.5 at 116-17 (1989). In short, *Baker v. Selden* consigns functional works to a regime of "thin" protection in order to

defend the line of demarcation between patent and copyright law.[13]

A weakening of *Baker v. Selden*'s principle of "thin" copyright protection for functional writings would run counter to the Court's recent decision in *Feist*, 499 U.S. 340. Relying in part on *Baker v. Selden*, the Court ruled that a competitor could copy commercially valuable but unprotectable facts from the plaintiff's directory. *Id.* at 350, *cf. Bonito Boats*, 489 U.S. 141 (state law preempted by patent law because it would have removed from the public domain functional designs that had not undergone the patent examination process). Moreover, *Feist* made clear that the rule of "thin" copyright protection is of constitutional stature. *Feist*, 499 U.S. at 349-50. In the case of functional writings, the constitutional authorization to enact legislation to promote progress in the useful arts by granting a limited term of exclusive rights to inventors underscores the constitutional appropriateness of limiting the scope of copyright protection available to works depicting content that might be patented.

[13] *Baker*, 101 U.S. at 102; *Kepner-Tregoe, Inc. v. Carabio*, 203 U.S.P.Q. (BNA) 124, 130 (E.D. Mich. 1979) (thin protection of functional works is an "open secret"); *Sega*, 977 F.2d at 1524 (discussing reasons that functional writings have a thinner scope of protection than other classes of works); *Altai*, 982 F.2d at 704 ("compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in § 102(b)"). *See also* Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 So. Cal. L. Rev. 1091 (1995); Mark A. Lemley, *Convergence in the Law of Software Copyright?*, 10 High Tech. L.J. 1 (1995).

C. Computer Programs Embody Many Useful Methods and Systems That Should Be Protected, If At All, By The Patent System.

Unlike traditional literary works, such as novels, computer programs typically embody or implement potentially patentable procedures, processes, systems and methods of operation.[14] Many thousands of patents have issued for program-related inventions in recent years, including utility patents for the following kinds of program components: (1) algorithms applied to industrial uses, *see, e.g.*, Patent No. 4,744,028 (algorithm for efficient resource application); (2) efficient data structures, *see, e.g., In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994); (3) user interface techniques, *see, e.g.*, Patent No. 5,467,448 (method for adjusting the format of tables intended for implementation in computer programs); (4) systems for program-to-program or program-to-machine interfaces, *see, e.g., Atari Games Corp. v. Nintendo of America, Inc.*, 30 U.S.P.Q.2d 1401 (N.D. Cal 1993) and (5) systems for controlling the operations of particular kinds of machines, *see, e.g., In re Abele*, 684 F.2d 902 (C.C.P.A. 1982) (improved method for controlling operations of CAT-scan machine).

Patents have issued for these program elements because the patent statute identifies processes and machines as patentable subject matter, 35 U.S.C. § 101. Patent claims frequently characterize the nature of the claimed invention by describing them as a system for doing a particular task or a method of operating a device to accomplish some task. It

[14] Of course, there are many aspects of programs that both copyright and patent law would regard as unprotectable "ideas." *See, e.g., In re Meyer*, 688 F.2d 789, 794 (C.C.P.A. 1982) (noting the Court's longstanding exclusion of "scientific principles, laws of nature, ideas, and mental processes" from the subject matter of patent law); 17 U.S.C. § 102(b) (copyright's exclusion of ideas and principles).

is no wonder, then—and no mere coincidence either—that 17 U.S.C. § 102(b) identifies "process[es], system[s], [and] method[s] of operation" as elements that when embodied in works of authorship are beyond the scope of copyright. Moreover, patents routinely recite prior methods or systems of performing the same function in distinguishing the claimed invention from the prior art. Because of this, the availability of alternative choices is not by itself a reliable basis for distinguishing between elements of a program that are expressive and those that are excludable under § 102(b).

In recent years, courts in copyright cases involving computer programs have become increasingly sensitive to the utilitarian nature of computer programs and the important role patents play in providing legal protection to computer program innovations. In *Atari Games Corp. v. Nintendo of America, Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992), for instance, the court stated that the author of a computer program should look to copyright law to protect the expression in her program, but to patent law to protect the processes or methods of operation that might be embodied in the program. *See also Sega*, 977 F.2d at 1526 and *Borland*, 49 F.3d at 819, Pet.App. 23a. (Boudin, J., concurring) (noting that extending a broad scope of copyright protection to computer programs "can have some of the consequences of *patent* protection in limiting other people's ability to perform a task in the most efficient manner").

D. The District Court Improperly Applied Copyright To Protect Useful Methods or Systems In The Lotus Program.

As noted above, there are some among us who are persuaded by the First Circuit's ruling that the Lotus command hierarchy is an unprotectable method of operating a computer to perform spreadsheet functions. However, in the event the Court is not persuaded by the "method of operation" rationale employed by the Court of Appeals, we wish

to bring to the Court's attention other troublesome aspects of the District Court's copyright analysis in this case.

First, the District Court characterized the Lotus command hierarchy as "a fundamental part of the functionality of keystroke sequences and the macro language." *Borland II*, 799 F. Supp. at 207, Pet.App. 110a; *see also id.* at 213, 219, Pet.App. 123a, 134a. It would appear to us that a macro language and fundamental parts of the functionality of a macro system would be beyond the scope of copyright protection under well-established principles deriving from *Baker v. Selden* and the "shorthand system" cases. *See, e.g., Brief English*, 48 F.2d 555 and *Griggs v. Perrin*, 49 F. 15 (C.C.N.D.N.Y. 1892).[15]

Second, the District Court seemed more bothered by functional uses of the Lotus command hierarchy by Lotus's competitors than by informative displays of them. *Compare Paperback*, 740 F. Supp. at 69, Pet.App. 237a, (suggesting it would be lawful for a competing spreadsheet product to display the Lotus commands on a help screen to inform users about the equivalent command in the second program, or to have a macro conversion facility such as that provided in Microsoft's Excel product) with *Borland II*, 799 F. Supp. at 219, Pet.App. 134a. ("[T]he macros and keystroke sequences are protected *to the extent that it is necessary to infringe a copyright to use them.*") (emphasis added). From the standpoint of traditional principles of copyright law, this strikes us as backwards.

Third, in its most recent *Borland* decision, the District Court found copyright infringement arising from Borland's "key reader" feature that permitted users to use the same keystrokes to perform the same functions as the Lotus

---

[15] *See also Samuelson, supra* note 12, at 334-35, n.109 (citing commentaries critical of concept that languages can be copyrighted).

program. Execution of program functions through use of this feature does not involve any display of Lotus commands or their hierarchy. Indeed, in this last decision, the District Court seems to have found infringement based on "originality" in the selection and arrangement of executable functions of a program. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223, 231 (D. Mass. 1993), Pet.App. 40a-41a (*Borland IV*).

Finding infringement based on similarities in executable functions comes perilously close to finding infringement based on the fact that the two programs perform the same functions. This is not what Congress intended when it enacted § 102(b).[16] Nor is it what Congress expected when enacting the computer program-related amendments to the copyright statute in 1980. Congress had been reassured by CONTU that as long as programmers wrote their own code, it should not infringe copyright for two programs to perform the same functions. CONTU Final Report at 21-22.

There is no question that copyright law provides protection to the literal code of computer programs, that is, to the set of statements and instructions that can be used in a computer to bring about certain result. 17 U.S.C. § 101. The courts have enforced this Congressional mandate, and the protection that copyright has provided to program code has incented considerable investments in software development. Copyright has also provided protection to some detailed elements of the internal structure of programs. *See, e.g., Altai*, 982 F.2d at 702-03. There is, however, nothing in the Copyright Act of 1976, its legislative history, or the computer program-related amendments to this Act added in

[16] *See, e.g.*, Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308, 2351 (1994) ("program behavior, in general, is unprotectable by copyright law on account of its functionality").

1980 to indicate a Congressional intent to extend copyright protection to the functional results occurring when program instructions are executed, such as when a program controls the operations of a nuclear power plant or performs spreadsheet functions.[17] These functional results are processes of the sort that Congress meant to exclude from the scope of copyright protection by enactment of section 102(b).

If computer programs need more protection against competitive imitation than copyright, supplemented by patent and trade secrecy law, can provide,[18] software developers should seek additional legal protection for computer programs from Congress. Even though incentive-based arguments for extending copyright law to reach all commercially valuable aspects of computer programs may have some appeal, adopting such a rule would ultimately have "a corrosive effect on certain fundamental tenets of copyright doctrine," as the Second Circuit so aptly noted in *Altai*, 982 F.2d at 712.

V. CONCLUSION

Computer programs have posed many vexing questions for copyright law, including the difficult issues presented by the present litigation. As the Second Circuit Court of Appeals noted in *Altai*: "Thus far, many of the decisions in this area reflect the courts' attempt to fit the proverbial square peg in a round hole." *Altai*, 982 F.2d at 712. Even so,

[17] Sometimes, of course, the results achieved by execution of program instructions will be textual (an electronic book, for example) or pictorial (such as videogame graphics). In such cases, courts will be able to apply traditional copyright principles when charges of infringement arise. *See, e.g., Atari, Inc. v. North American Philips Consumer Elec. Corp.*, 672 F.2d 607 (7th Cir. 1982) (finding infringement because of similarities in pictorial details of two videogames).

[18] *See, e.g.*, Samuelson, et al., *supra* note 16, at 2342-61.

courts have been making progress in using case-by-case analysis to develop criteria for distinguishing the expression in programs from the methods or processes that are unprotectable by copyright law under § 102(b), just as Congress intended. The utilitarian nature of computer programs necessarily means that they will enjoy a narrower scope of copyright protection than artistic and fanciful works. Useful methods and processes embodied in or carried out by computer programs should be regulated by patent law, not copyright. As with the exclusion of facts from the scope of copyright by virtue of § 102(b), the exclusion of useful methods and processes from the scope of copyright "is not 'some unforeseen byproduct of some statutory scheme . . . ,' " but "rather, 'the essence of copyright,' and a constitutional mandate." *Feist*, 499 U.S. at 349 (citation omitted).

Respectfully submitted,

PAMELA SAMUELSON
(Counsel of Record)
CORNELL LAW SCHOOL
322 Myron Taylor Hall
Ithaca, NY 14853
(607) 255-2829

Attorney for Amici
Copyright Law Professors

December 1995

(13)

No. 94-2003

Supreme Court, U.S.
FILED
DEC 7 1995
CLERK

In the Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
PETITIONER,

v.

BORLAND INTERNATIONAL, INC.,
RESPONDENT.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

BRIEF AMICUS CURIAE OF COMPUTER SCIENTISTS IN SUPPORT OF RESPONDENT

RON KILGARD
(Counsel of Record)
KARL M. TILLEMAN

Dalton, Gotto, Samson & Kilgard, P.L.C.
Suite 900
3101 North Central
Phoenix, AZ 85012
Phone: (602) 248-0088

Attorneys for Amici

December 1995

BEST AVAILABLE COPY

4 1PM

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
I. INTRODUCTION AND INTEREST OF AMICI	1
II. SUMMARY OF THE ARGUMENT	3
III. ARGUMENT	4
A. General Discussion of Computers	4
B. Human/Computer Interfaces Should Not Be Copyrightable	6
1. Overview of Human/Computer Interfaces	6
2. Reasons Why Copyright Protection Is Not Appropriate	8
C. Computer Languages Should Not Be Copyrightable	11
1. Overview of Computer Languages	11
2. The Lotus Macro Language	13
3. Reasons Why Copyright Protection Is Not Appropriate	16
D. The Menu Command Words Need Not Be Spelled Out in the Code	18
IV. CONCLUSION	19

TABLE OF AUTHORITIES

Page(s)

CASES

Bonito Boats, Inc. v. *Thunder Craft Boats, Inc.*, 489 U.S. 141, 103 L. Ed. 2d 118, 109 S.Ct. 971 (1989) 10

Computer Associates Intern., Inc. v. Altai, Inc., 775 F.Supp. 544 (E.D.N.Y. 1991) 4, 5

Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 10

Fogerty v. Fantasy, Inc., 510 U.S. __, 127 L.Ed.2d 455, 114 S.Ct. 1023 (1994) 10

Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807 (1st Cir. 1995)..... 6, 7, 8, 13, 17

Lotus Development Corp. v. Borland International, Inc., 778 F.Supp. 78 (D. Mass. 1992) 13

Lotus Development Corp. v. Borland International, Inc., 799 F.Supp. 203 (D. Mass. 1992) 13, 16

Lotus Development Corp. v. Borland International, Inc., 831 F.Supp. 202 (D. Mass. 1993) 13

Lotus Development Corp. v. Borland International, Inc., 831 F.Supp. 223 (D. Mass. 1993) 13, 16

Lotus Development Corp. v. Paperback Software International, 740 F.Supp. 37 (D. Mass. 1990) 13, 16

STATUTES

17 U.S.C. § 101 ..... passim

17 U.S.C. §106(2) 18

MISCELLANEOUS

Compendium II of Copyright Office Practices, the Library of Congress, §326 at 300-32 (1984) 11

Paul A. David, "CLIO and the Economics of QWERTY," 75 *Am. Econ. Rev.* 332 (1985) 7

E. Lowry, *Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?*, 39 Emory L.J. 1293 (1990) 11, 16

F. Friedman, E. Koffman, Problem Solving, Abstraction, and Design Using C++, at 6-18 (Addison-Wesley Publishing Co. 1994) 4

Paul Goldstein, *Copyright Principles, Law and Practice*, §1.2.2.4 at 16 (1989) 16

IBM DICTIONARY OF COMPUTING (McGraw-Hill 10th Ed. 1993) 4, 6

P. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stanford L. Rev. 1045 (1989) 10

MICROSOFT PRESS COMPUTER DICTIONARY (2d Ed. 1994) 5, 6

Richard A. Shaffer, *Computers With Ears*, Forbes, September 12, 1994, at 238 8

R. Stern, *Copyright in Computer Programming Languages*, 17 Rutgers Computer & Tech. L.J. 321 (1991) 11, 16

Wayne Rash, Jr., *Talk Show: Voice-recognition Technology for the PC Lets Your Voice Run the Show*, PC Magazine, December 20, 1994, at 203 8

Users Guide for WordPerfect for Windows 6.1 at 305-308 (Novell, Inc. 1994) 7

THIS PAGE INTENTIONALLY LEFT BLANK

No. 94-2003

In the Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
PETITIONER,

v.

BORLAND INTERNATIONAL, INC.,
RESPONDENT.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FIRST CIRCUIT

BRIEF AMICUS CURIAE OF COMPUTER SCIENTISTS IN SUPPORT OF RESPONDENT

I. INTRODUCTION AND INTEREST OF AMICI

This amicus brief is filed on behalf of numerous individual computer scientists who believe the opinion of the U.S. Court of Appeals for the First Circuit was correctly decided and should be upheld. The signatories to this brief include some of the leading, pioneering scientists in the computer industry. We have invented or contributed to the authorship of numerous computer programs and languages.[1] We have prepared this

[1] A full list of amici and their professional qualifications and contributions to the industry follows the signature page of this brief. Amici include computer scientists who developed or contributed to the development of

brief because we believe the First Circuit wisely corrected the District Court's error in overextending copyright coverage in a manner irreconcilable with the purpose of Copyright law and the nature of computer science. This brief was approved by each of the individual signatories who exercised complete control over its editorial contents. At our request, Borland helped defer the costs of printing. Both parties have consented to the filing of this brief.

Many of us filed an *amicus* brief in the First Circuit Court of Appeals, in which we discussed the copyrightability of computer languages, and urged reversal of the District Court's rulings. The District Court's opinions had generated widespread controversy and uncertainty in the computer industry because they went far beyond earlier case law, which recognized copyright coverage for the text of a computer program ("code") or for the visual, pictorial displays of the computer screen, but not for "computer processes." The District Court, however, held that functional aspects of Lotus' 1-2-3 spreadsheet program, and the program's procedures and methods of operation, were copyrightable and infringed, even though Borland copied neither the spreadsheet's programming code nor its visual displays. Specifically, the District Court held as copyrightable the program's menu commands, which are used to operate the computer, and which are used as a "macro language" for programming computer routines to execute a desired computer operation repeatedly. In protecting these computer processes, the District Court ignored the distinction between an expressive work (such as a novel or a computer program's "code"), on the one hand, and the language or medium used to express a work (such as the English language or a computer programming language), on the other hand. The First Circuit remedied that legal error by holding the menu command hierarchy to be an uncopyrightable method of operating the computer.

Not only does the First Circuit's opinion properly interpret the copyright laws, it also makes sense from a practical point of view. Fundamentally, this case directly affects the way people

(1) computer languages such as Ada, Lisp and Logo; (2) artificial intelligence; (3) robotics; (4) data encryption; and (5) other important areas of computer science.

operate machines such as computers, and the way people communicate with computers using programming languages. This *amicus* brief explains two ways in which a user or programmer would use the menu command words of Lotus 1-2-3, namely (1) as part of the human/computer interface to run the program, just as buttons or switches were historically used to run machines, and (2) as a computer language to write macro programs. Such standardized computer interfaces and computer languages are highly important. Progress in the computer industry depends upon software compatibility and the ability of computer programmers to communicate freely with each other, using common computer interfaces and computer languages. Affirmance of the First Circuit's decision will help further these important policy goals. Because the copyrightability of computer interfaces and languages is a crucial issue to the computer industry, we respectfully submit the following brief on why computer interfaces, such as the Lotus 1-2-3 menu command hierarchy and macro language, should not be copyrightable.

II. SUMMARY OF THE ARGUMENT

1. Profound historical changes have occurred over the last several decades in the way in which people communicate with machines. Buttons, switches, and electrical plugs have been replaced with computers, and a variety of human/computer interfaces. One such interface consists of the Lotus 1-2-3 menu command words and the sequencing or syntax of those words, or "menu command hierarchy." Lotus uses menu words such as "COPY" or "PRINT" in place of buttons or switches which accomplished analogous functions on earlier machines.

2. Buttons, switches, and plugs have never been considered copyrightable on their own. Replacing them with their computer equivalents should make no difference under the law. This conclusion holds true even though a "computer program" is involved. Computer interfaces or languages are not the same as the computer program itself, which is the only aspect of a computer Congress has said should be copyrightable. Instead, a computer interface is an uncopyrightable method of operating a computer, as the First Circuit correctly found here.

3. The First Circuit's opinion also recognizes the reality faced by the computer industry. Standardization of computer interfaces and languages is vitally important to this industry. Requiring each new programmer to write a new interface or language would be absurdly inefficient and would decrease, not increase, the number of creative works in the industry.

4. In this case, the Lotus 1-2-3 menu command words and the sequencing or syntax of those words also define a computer language in which users can write computer programs known as "macros." As a computer language, the Lotus menu words and order are not the same thing as a computer program. It is very important to distinguish a copyrightable computer program from the uncopyrightable language in which the program is written. Languages are the building blocks used to create works such as novels or computer programs; protecting those building blocks is quite different from protecting the expression in a work created with those building blocks. Protecting a novel written in the English language is not the same as protecting the English language itself. Before this case, no one would have seriously argued that any language could be copyrightable, let alone a computer language. There are a number of important policy reasons, discussed below, why computer languages should remain uncopyrightable.

III. ARGUMENT

A. General Discussion of Computers

The following is a basic discussion of computers helpful to understanding the issues discussed in this brief. Broadly speaking, computers have two components: *hardware* and *software*.[2] Hardware consists of the physical components used to make a

[2] Sources for this background material include: *Computer Associates Intern., Inc.* v. *Altai, Inc.*, 775 F.Supp. 544, 549-551 (E.D.N.Y. 1991), *aff'd.*, 982 F.2d 693 (2d Cir. 1992); F. Friedman, E. Koffman, Problem Solving, Abstraction, and Design Using C++, at 6-18, (Addison-Wesley Publishing Co. 1994), ISBN 0-201-52649-2; IBM Dictionary of Computing (McGraw-Hill 10th Ed. 1993), ISBN 0-07-031488-8(HC) (hereinafter *IBM Dictionary*); Microsoft Press Computer Dictionary, (2d Ed. 1994), ISBN 1-55615-597-2 (hereinafter *Microsoft Dictionary*).

computer, and includes three basic elements: the *central processing unit* ("CPU"), memory, and the input/output system. The CPU is the "brains" of the computer where the actual computing is done. On modern personal computers, the CPU is contained within a single silicon "chip" called a microprocessor. The CPU accesses various portions of the computer's memory to store and help run computer programs.

Software generally refers to computer programs, which are defined in 17 U.S.C. §101 as "a set of statements or instructions to be used directly or indirectly in a computer to bring about a certain result." Broadly speaking, there are two types of software: operating systems software and application programs. The *operating system software* is the basic set of instructions that allows the CPU to function as a computer, manages the interaction among the various hardware elements of the computer and between the hardware and the application programs, and allocates the computer's resources among other programs that might need them. The computer cannot function without operating system software. Because the operating system software must interact with other programs and with hardware components, the other programs must be compatible with the computer's operating system software, that is, they must be able to exchange information with the operating system in a precise and accurate manner. *See, e.g., Computer Associates*, 775 F.Supp. at 550. Examples of operating systems are DOS, Windows 95, and UNIX.

Once a computer has an operating system installed, it can accept additional computer programs, called *applications programs*, which perform applications such as word processing, payroll and inventory management systems, graphics, video games, or, as in this case, computer spreadsheets.

When human beings write computer programs, they generally do so in human readable code, known as *source code*. Source code is written in computer languages such as assembly

language, BASIC, or C++. In order for the computer to understand a program, a source code program must be translated into the "1s" and "0s" which are understandable to computers. This is called *object code*, and is virtually impossible for most humans to read.

B. Human/Computer Interfaces Should Not Be Copyrightable

1. Overview of Human/Computer Interfaces

This brief now explains the two ways in which a person running the Lotus 1-2-3 spreadsheet uses the Lotus menu command words, and their order. First, the person running the Lotus program uses the menu words to operate the spreadsheet. Therefore, the words and syntax are part of what is called the *human/computer interface,* or *user interface*. Generally speaking, an *interface* is a shared boundary, or the point at which a connection is made between two different elements or functional units in a computer system so that they can work with one another. *See Microsoft Dictionary* at 218; *IBM Dictionary* at 351. The user interface consists of the graphical design, the commands, prompts, and other devices that enable a user to interact with and perform operations on a computer system or program. *See Microsoft Dictionary* at 218; *IBM Dictionary* at 724. The Lotus menu commands meet these definitions, as they are directions to the computer to "manipulate and control the program." *Lotus Development Corp.* v. *Borland International, Inc.*, 49 F.3d 807, 809 (1st Cir. 1995), Pet. App. at 4a ("*Lotus* v. *Borland*"). More specifically, the Lotus menu command words and syntax are the "specification" for the human/computer interface. This means that the human operator must use the exact words in their exact order to operate the Lotus spreadsheet. No other words or order will do.

On a steadily increasing basis, human interaction with machines is now accomplished by human/computer interfaces. The First Circuit's analogy to the video cassette recorder (VCR) is one example. *Lotus* v. *Borland*, 49 F.3d at 817, Pet. App. at 18a. Historically, users pressed buttons to operate electronic devices such as VCRs. The first VCRs were not computer controlled; pressing the buttons operated electronic circuitry which in turn performed record, play, and other functions. Timers and clocks were largely mechanical. Modern VCRs have replaced those electrical and mechanical components with a computer. The user can still operate many of the VCR functions by pressing buttons that still say "record" or "play." Thus, the computer/human interface *looks* the same as it did 20 years ago; the only difference is that the buttons now operate a computer instead of electrical or mechanical components.

Similarly, the manual typewriter has been replaced by a computer keyboard. It still uses the same QWERTY computer/human interface, even though that interface was designed to be used with the relatively inefficient manual typewriter and is far from the optimum design for a computer keyboard. Paul A. David, "CLIO and the Economics of QWERTY," 75 *Am. Econ. Rev.* 332 (1985). Nor are contemporary computer keyboards just a collection of hardware buttons. Modern word processing programs typically have features permitting the keyboards to be redefined, in software, so that the keys can represent different letters or functions than they normally do. *See, e.g., User's Guide for WordPerfect for Windows 6.1* at 305-308 (Novell, Inc. 1994) (discussing "Keyboard Preferences" and "Keyboard Editor" features).

Computers also have progressed to the point where software "buttons" have replaced hardware buttons or switches. Lotus 1-2-3 is such an example; commands such as "COPY" or "PRINT" appear on the screen, instead of a physical button. *Lotus* v. *Borland*, 49 F.3d at 817, Pet. App. at 18a-19a. Other programs use "touch screen" menus where the user can touch a word displayed on a computer screen and operate the desired function.

Moreover, technology has progressed to the point where no manual interaction between a human being and a computer is

even necessary to operate the interface. At least three companies, including Lotus' owner, IBM Corp., are selling voice dictation and voice command systems enabling the user to speak the commands to the computer instead of either typing them, using function keys, or using a mouse[3].

2. Reasons Why Copyright Protection Is Not Appropriate

Twenty years ago, a company such as Lotus could not stop a competitor from using the organization of the buttons and switches on its product merely by paying $20 and filing a two-page form with the Library of Congress. Today, it is very difficult to see why Lotus should be allowed to use the copyright laws to stop its competitors from using the same command words of the Lotus program in the same order, when those words perform precisely the same function as a set of buttons did twenty years ago. The First Circuit noted that hardware buttons would not be copyrightable simply because of the "useful article" exception in the definitions of the Copyright Act, 17 U.S.C. § 101. *Lotus* v. *Borland*, 49 F.3d at 817, Pet. App. at 19a. As scientists, it is extremely difficult to understand why replacing hardware buttons with software buttons logically should change a formerly uncopyrightable "useful article" into something that is copyrightable, just because a "computer program" is involved. A human/computer interface, or a computer language, cannot be viewed in the same manner as any ordinary "literary work" exempt from the useful article exception of § 101. Someone can read an ordinary literary work, such as a novel or a play, and enjoy it for what it is worth. Someone can also read the text of a computer program written in source code as a work of authorship and gain some technical knowledge in the same way as one might gain such knowledge by reading a book about computers.

---

[3] In addition to IBM, the companies include at least Dragon Systems, Inc. of Newton, Massachusetts, and Kurzweil Applied Intelligence, Inc. of Waltham, Massachusetts. *See* Wayne Rash, Jr., *Talk Show: Voice-recognition Technology for the PC Lets Your Voice Run the Show*, PC Magazine, December 20, 1994, at 203, 205; Richard A. Shaffer, *Computers With Ears*, Forbes, September 12, 1994, at 238.

Computer interfaces are fundamentally different from any of these. They are singularly functional and utilitarian in nature. One does not normally "read" a set of commands such as "COPY" or "PRINT," or the syntax or structure of those commands, as one might read a book about computers. Rather, the purpose of human/computer interfaces is to operate functions of the computer machine itself, and to use the computer.

Because of the nature of human/computer interfaces, there are a number of policy reasons why they should not be the subject of copyright protection. First, factors such as software compatibility, interoperability, and hardware portability are extremely important in the computer industry. Computers essentially have replaced the methods we used to operate machines in the past (hardware buttons or switches) with identically functioning buttons that instead operate software. In some instances, the physical switches appear the same to the user; in other instances, they are words on a computer screen.

This trend will only accelerate in the future. With computing hardware becoming increasingly inexpensive, software "virtual machines" have replaced electronic hardware in entire sectors of the economy. (A software "virtual" machine is one where the functions of electronic or mechanical components are primarily performed by software.) The ramifications of this computer revolution are enormous. The number of different kinds of computer/human interfaces have dramatically increased, both in sheer quantity as well as in the many different ways in which humans now interact with computers on a daily basis. Voice commands, and doubtless other non-physical methods of communication not presently imaginable, will increasingly replace hardware and even software buttons and switches.

The ability of programmers to write software programs that can be understood by other computers worldwide, on many types of computer platforms, is crucially important. The key point is that once an interface is created, it can become an industry standard, and the interface becomes functionally embedded not just in the original work, but in programs, and in

the minds of programmers. The QWERTY typewriter keyboard and the positioning of automobile controls such as the gas and brake pedals and gear shift patterns are hardware examples of such standards; common computer languages such as BASIC and C++ are examples in the software industry.

Once a standard interface has been created, it would be absurdly inefficient to require all subsequent authors of computer software to create their own interface with each new program. Imagine if each author of a novel were required to invent a new language because a prior author had a copyright in the English language. This would lead to *less* works of authorship being created, rather than encouraging the proliferation of works as the copyright laws were intended to do. *See Fogerty* v. *Fantasy, Inc.*, 510 U.S. ___, 127 L.Ed.2d 455, 465, 114 S.Ct. 1023, 1029 (1994) (noting the ultimate aim of the Copyright Act is "to stimulate artistic creativity for the general public good").

Thus, granting copyright protection in computer interfaces would allow the creator or "first comer" to create a computer interface and use it to lock up the technology and marketplace for 75 years, even if the interface and technology was unpatentable. This would create enormous barriers to entry in a market where the free exchange of ideas is essential. *See, e.g. Computer Associates International, Inc.* v. *Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992), citing P. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stanford L. Rev. 1045, 1082 (1989). This effective monopoly on a computer interface would inhibit third-party innovations, particularly given the long term of copyright. *Cf. Bonito Boats, Inc.* v. *Thunder Craft Boats, Inc.*, 489 U.S. 141, 146, 103 L. Ed. 2d 118, 131, 109 S.Ct. 971, 975 (1989) ("imitation and refinement through imitation are both necessary to invention itself and the very lifeblood of a competitive economy."). If the interface was copyrightable, the interface owner also would "lock up" an installed base of users accustomed to that interface.

C. Computer Languages Should Not Be Copyrightable

1. Overview of Computer Languages

This portion of the brief discusses the second way a user uses the Lotus menu words, as a computer language to write macro programs.[4] The Copyright Office has defined a computer language as "a programming language used by a programmer for writing a computer program."[5] A commentator similarly has described a computer language as "a code that both the computer operator and the computer understand. The operator uses the language to express what is to be done, and the computer understands the language and performs the desired actions."[6] A computer language thus is similar in many respects to an ordinary language such as English. Like an ordinary language, it consists of basic building blocks of communication. It provides rules (grammar and syntax) for the combination of words or terms to form expressive works that can be understood by their authors and other programmers.

---

[4] Lotus has denominated the Lotus macro language as the method by which the macro programs can interact with the Lotus 1-2-3 program. *See* Declaration of Vern L. Raburn, ¶ 18 and Exhibit B, J.A. 530, 535. In this respect the Lotus macro language acts as a program-to-program interface specification. We understand that other *amici* are discussing the uncopyrightability of program-to-program interfaces, and therefore will not discuss it in this brief.

[5] Compendium II of Copyright Office Practices, the Library of Congress, §326 at 300-32 (1984).

[6] E. Lowry, *Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?*, 39 Emory L.J. 1293, 1298 (1990) (citation omitted) (hereinafter "*Lowry*"). Another commentator summarized a number of definitions of computer language as follows:

> A computer programming language is a formal system of expression including:
> (1) a set of vocabulary elements;
> (2) a set of syntax rules for combining vocabulary elements into statements; and
> (3) the assignment of meaning to statements that properly combine vocabulary elements in accordance with the syntax rules.

R. Stern, *Copyright in Computer Programming Languages*, 17 Rutgers Computer & Tech. L.J. 321, 327 (1991) (hereinafter "*Stern*").

Unlike ordinary languages, computer languages serve an important additional purpose. They not only permit communication between different computer programmers, but also allow programmers to communicate with a computer and tell it what to do. They are highly utilitarian languages that tell the computer how to implement the desired functions, operations, and results desired by the program. In short, they permit computers to "talk" to each other and to exchange that which will be understood by the operators of the computer.

A computer language is fundamentally different than a program written in that language. The rules for the language consist of definitions and syntax, usually written in alphabetical or some other logical order. However, a program uses selected words of the language in the order appropriate for the function being performed. This concept is illustrated by the following example using the BASIC programming language. (The commands are listed in the alphabetical order used in the BASIC manual, and are only the small group of the many available BASIC commands which are actually used in the sample program, plus a few others.)

COMPARISON OF BASIC COMMANDS
WITH A PROGRAM WRITTEN IN BASIC

BASIC Commands	Sample Computer Program
DATA	10 PRINT ''This program adds.''
END	20 PRINT ''Type a number.''
GOTO	30 INPUT A
IF/THEN	40 PRINT ''Type another number.''
INPUT	50 INPUT B
LET	60 LET C = A + B
PRINT	70 PRINT ''The sum is'', C
READ	80 END

The above program asks the user to type in two numbers; it then computes the sum of the two numbers and prints the sum. In the terminology of 17 U.S.C. § 101, the eight lines numbered

10 through 80 are the "set of statements or instructions" of the program; the printed sum is the "certain result" of the program. Thus, the program on the right is different from the list of commands on the left. (While the "certain result" in this example is a number, in other programs the certain result can be other matters such as a still or video image, audio recordings, or, as in this case, command words such as "COPY" or "PRINT").

2. The Lotus Macro Language

In this case, the Lotus 1-2-3 menu command words and the sequencing or syntax of those words define a computer language. Lotus 1-2-3 allows users to write their own computer programs, called "macros," using this programming language. *See Lotus* v. *Borland*, 49 F.3d at 809, Pet. App. at 4a. The District Court defined a "macro language" as "a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence."[7] As explained by the District Court, the Lotus 1-2-3 macro language permits a user to write a series of executable computer operations (the "macros") by using a language of special abbreviated programming commands (such as "R", "F", or "C"), and special symbols (such as "/" or "{?}"). *Borland IV*, 831 F.Supp. at 226-27, Pet. App. at 31a-32a. Thus, the 1-2-3 macro language is a set of rules for writing a computer program (the macro), and meets the definition of a computer language discussed above. Once users become reasonably adept with the operation of the spreadsheet program, they will often write macros to save time and effort.

---

[7] *See Lotus Development Corp.* v. *Borland International, Inc.*, 799 F.Supp. 203, 206 (D.Mass. 1992), Pet. App. at 110a. (hereinafter "*Borland II*"). This brief also refers to four other relevant opinions by the District Court: *Lotus Development Corp.* v. *Paperback Software International*, 740 F.Supp. 37 (D.Mass. 1990) (hereinafter "*Paperback*"); *Lotus Development Corp.* v. *Borland International, Inc.*, 778 F.Supp. 78 (D.Mass. 1992) (hereinafter "*Borland I*"); *Lotus Development Corp.* v. *Borland International, Inc.*, 831 F.Supp. 202 (D.Mass. 1993) (hereinafter "*Borland III*"); *Lotus Development Corp.* v. *Borland International, Inc.*, 831 F.Supp. 223 (D.Mass. 1993) (hereinafter "*Borland IV*").

Lotus has argued that the menu command hierarchy is a "computer program," and hence protectible under the copyright laws. For the reasons discussed above, this is not so. The words and syntax of the hierarchy can be used to write copyrightable computer programs, *i.e.* the macros. However, when the Lotus menu words and syntax are used in this manner, they are a language, not a program. They are the words from which the macro programmer creates the macros, which are the "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result" under 17 U.S.C. §101. In this respect, the Lotus menu words and syntax are similar to the words and syntax of a "national language," like English or French, which can be used to create copyrightable works of literature, and are similar to the BASIC language example noted above.

Again, this can be illustrated by an example comparing some of the Lotus command words with a macro written using those words. In the following example, the left column lists the "File" and "Data" commands used in the first two lines of the sample macro, in the order in which they appear in Lotus 1-2-3, as shown at J.A. 900-906. The macro example on the right is taken from the Declaration of Judith S. Olson, J.A. 508-09.

COMPARISON OF LOTUS 1-2-3 COMMANDS WITH A LOTUS MACRO PROGRAM

Lotus Commands	Sample Macro
''/F''ile Commands	/fit{?}~
. . .	/dprfcfe{?}~
''I''mport	i.{end}{down}~
''T''ext	o{right}~g
. . .	
''/D''ata Commands	
. . .	
''P''arse	
''F''ormat Line	
''C''reate	
''E''dit	
. . .	
''R''eset	

As Dr. Olson's declaration explains (J.A. 508-09), the macro program is used to extract data from another file, such as a database, and re-format it for use in a spreadsheet. The first line of the macro, for instance, opens a file ("f"), and imports it ("i") from the database as text ("t"). (Unlike BASIC, which as shown in the previous example uses entire English-looking words such as "PRINT," the Lotus macro language often uses just the first letter of the relevant command. *See Borland IV*, 831 F.Supp. at 226-27, Pet. App. at 31a.) The four lines of the macro are the "set of statements or instructions" of the program; the re-formatted data file is the "certain result" of the program. As was the case with the previous example, the program on the right is different from the list of commands on the left.

3. Reasons Why Copyright Protection Is Not Appropriate

It is very important to distinguish a copyrightable computer program from the uncopyrightable language in which the program is written. Copyright law was intended to cover the expressive nature of an author's works, and to encourage their creation. However, it was never meant to cover the basic building blocks, or language, used to create the works.[8] Protecting the expression in a work is quite different from protecting the building blocks used to create the work. Protecting a novel written in the English language is not the same as protecting the English language itself. Nor is protecting a work of art such as a painting the same as protecting the various colors of paint used to create the painting. As the above examples make clear, a computer language is not the same as the "computer program" defined in 17 U.S.C. § 101. Accordingly, before this case, no one would have seriously argued that any language could be copyrightable, let alone a computer language.[9]

There also are a number of good policy reasons why computer languages should not be copyrightable. The reasons discussed above why copyright protection is not appropriate for human/computer interfaces generally also apply to computer languages. Standard languages are very important for ε wide variety of reasons: promoting ease of use by programmers, avoiding lock-ins of the user base and the technology by the

---

[8] *Lowry* at 1315; I. Paul Goldstein, *Copyright Principles, Law and Practice*, §1.2.2.4 at 16 (1989). See also *Stern* at 349, 364.

[9] The District Court implicitly granted copyright protection to the Lotus menu command hierarchy as a computer language. *Borland IV*, 831 F.Supp. at 229-230, 234, Pet. App. at 37a-39a, 47a; see also *Borland II*, 799 F.Supp. at 213-14, Pet. App. at 122a-125a. The District Court seemed to think that languages generally were copyrightable. *Paperback*, 740 F.Supp. at 72, Pet. App. at 243a-244a. See *Stern* at 323-24, 330, noting that *Paperback's* statement to this effect was unprecedented. This was one of the reasons why the District Court's opinions generated such controversy in the computer industry and why a number of us filed an *amicus* brief in the First Circuit criticizing the District Court.

first company on the market to write a language, providing incentives to create new works using a standard language, and preventing the disincentives that arise from requiring each competitor to create its own language.

The First Circuit acknowledged the realities faced by the computer industry. The majority opinion recognized the importance of program compatibility and the importance of allowing users to run programs such as macros on different platforms. The majority opinion concluded that "forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in §102(b) that 'methods of operation' are not copyrightable." *Lotus* v. *Borland*, 49 F.3d at 817-18, Pet. App. at 20a. Judge Boudin's concurring opinion also correctly observed how providing a monopoly of a computer command structure would make the users "captives" of the first provider of the program, not because of any creative effort or investment by that programmer, but because of the "investment in learning made by the users." *Id.* at 821, Pet. App. at 26a-27a.

Providing copyright protection for a computer language, such as the Lotus macro language, would have an additional undesirable effect. Such protection would extend far past the computer language itself. For example, granting copyright coverage for computer languages would allow the language owners to claim ownership of all programs written in the languages as derivative works. If languages were copyrightable, works created using the language would be derivative works of the language.[10] In that event, the computer language owner could

---

[10] *See* the definition of derivative work in 17 U.S.C. §101:

A 'derivative work' is a work based upon one or more preexisting works, such as a translation, . . . abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations or other modifications . . . is a 'derivative work'.

Under this statute, a program is a work "based upon" the preexisting language; the language is "recast, transformed or adapted" into the program, which in any event is an "elaboration or other modification" of the language.

argue that he owned all programs based on his language.[11] This objectionable result is a natural consequence of confusing the expressive work itself with the language used in creating the expressive work, and permitting copyright coverage for both.

D. The Menu Command Words Need Not Be Spelled Out in the Code

Finally, as computer scientists we wish to respond to one point in the Lotus brief which we believe to be incorrect or potentially misleading. Lotus states that the actual menu command words are "spelled out, in text, in the program code." Lotus Br. at 7. While the record cites at this point are to a *Borland* data file which needed to recognize the 1-2-3 command words and order to operate, Lotus' statement incorrectly implies that the words did in fact appear in the *Lotus* code. Since the Lotus code was never introduced into evidence, J.A. 300, one cannot conclude that the menu words actually appeared in the Lotus code. From a programming standpoint, there are a number of ways the Lotus code could have implemented the menu functions without spelling out the words in the code.

For example, an extremely common programming technique used for programs with international sales (such as word processors or database programs) does not spell out the menu command words in the program code. The menu words are instead contained in a data file which is separate from the code used to run the program. This makes the menu words easy to change when the program is sold to non-English speaking users, since the words are isolated in one location instead of being scattered throughout thousands or millions of lines of code. It also has the advantage of permitting users to customize the menu command words if they want to. While we do not know if this or a similar technique was actually used in Lotus 1-2-3, since the Lotus code was never placed into evidence it cannot

[11] *See* 17 U.S.C. §106(2).

be assumed that such a technique was *not* used. Thus, one cannot state, as does Lotus, that its menu words were "spelled out" in its code.

IV. CONCLUSION

As computer scientists, we believe that the majority and concurring opinions of the First Circuit correctly recognized the danger in extending copyright protection to the specific type of human/computer interfaces and computer languages at issue in this case. For people in our computerized society, it is vitally important to be able to communicate freely with computers, without threat of litigation. Computer interfaces and languages are meant for everyone to use. They were never meant to enable their creator to monopolize methods of operating computer-based machines. The District Court's decisions lost sight of the importance of being able to use common, standard computer interfaces. The First Circuit correctly understood the issues involved here, and remedied the District Court's error.

For the foregoing reasons, the judgment of the First Circuit should be affirmed.

Respectfully submitted,

RON KILGARD
(Counsel of Record)
KARL M. TILLEMAN

Dalton, Gotto, Samson & Kilgard, P.L.C.
Suite 900
3101 North Central
Phoenix, AZ 85012
Phone: (602) 248-0088

Attorneys for Amici
Listed in Appendix A

December 1995

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A
LIST OF SIGNATORIES TO THE BRIEF

[In Alphabetical Order]

1. Harold Abelson

Harold Abelson is Class of 1922 Professor of Computer Science and Engineering in the Department of Electrical Engineering and Computer Science at MIT. He was recipient in 1992 of the MIT Bose Award (School of Engineering teaching award) and is a Fellow of the IEEE. In 1992, Abelson was designated as one of MIT's six inaugural MacVicar Faculty Fellows, in recognition of his significant and sustained contributions to teaching and undergraduate education. He has made many contributions to the development of the Logo educational computing language. He directed the first implementation of Logo for the Apple II, which made the language widely available on personal computers beginning in 1981. Together with Prof. Gerald Sussman, Abelson developed MIT's introductory computer science subject, "Structure and Interpretation of Computer Programs," a subject organized around the notion that a computer language is primarily a formal medium for expressing ideas about methodology, rather than just a way to get a computer to perform operations. This work has had a world-wide impact on university computer-science education.

2. William B. Ackerman

Designer/implementor of one of the first capability-based multiprocessing operating systems, 1968; Author of the extremely popular ISPELL spelling checker, 1978; Co-author of the VAL dataflow programming language, 1982; Co-discoverer of the "Brock-Ackerman Anomaly" in denotational semantics of nondeterministic message-passing systems, 1984.

Currently at Hewlett-Packard Laboratories, Cambridge, MA.

3. Dean Anderson

I am President of the League for Programming Freedom, an organization founded to fight user interface copyrights and software patents. My software background includes 9 years of programming and system administration experience working on a wide variety of computer systems, languages, and operating systems for companies such as Charles Stark Draper Laboratory, Open Software Foundation, Kendall Square Research, and Hitachi Computer Products. I have contributed to the development of OSF/Motif, OSF/1 and other projects on systems ranging from mainframes to PC's. I have worked with a variety of operating systems from many flavors of unix to IBM MVS and VM to Microsoft DOS, Windows, and NT. I have programmed in a variety of languages from scheme, C, PLI, to TCL and Perl. In the last 4 years, I have worked on implementing a wide variety of networks including FDDI, CDDI, 100BaseT, Token Ring, Ethernet, Frame Relay, Dialup, supporting several protocols such as IP, Novell, and Appletalk. Currently, I am also President of Plain Aviation, Inc., a company started in 1992 which offers aviation and engineering services filling in corporate infrastructure gaps and recently offering Systems and Network Consulting, Custom solutions, and Co-sourcing services.

4. Judy Anderson
Harlequin Inc.
Cambridge MA 02142

Judy Anderson has been working as a programmer in the computer industry for over a decade. Her work has primarily been programming in Lisp and C, particularly doing low level implementation of Lisp programming environments. Of late, she has been interfacing between the Lisp system sold by Harlequin and various database products sold by other vendors.

5. Phil Bagley

Graduate of MIT in Electrical Engineering, Master of Science 1951. Thesis: High-speed digital machines for information searching. MIT Digital Computer Lab and MIT Lincoln Lab, 1951-1959, research associate. MITRE Corporation, 1959-1964, technical staff member.

Assistant Professor of Computers and Information Science, Temple University, 1978-1984.

6. Brian Bartholomew

I am a professional systems administrator for Unix, Windows, and Macintosh computers. I make my living extending the abilities of users by connecting and integrating disparate applications and operating systems in ways their manufacturers did not plan for.

7. Karl Berry

I am a member of the technical staff at Interleaf Inc., an Adjunct Research Associate at the University of Massachusetts at Boston, co-author or co-editor of several technical books published by Addison-Wesley and Kluwer, and maintain various freely available software packages.

8. Ethan Bradford

Education: BS, Mathematical Sciences, Stanford U., June 1981 (with Honors and Distinction); MS, Computer Science, Stanford U., June 1982; MS, Physics, U. of Washington, June 1991; PhD, Physics, U. of Washington, expected June 1992

Full time employment: Full Staff Member (doing programming and systems analysis), MIT Lincoln Laboratory, June 1982-August 1989

9. Seth Breidbart
Ph.D., Computer Science, Yale 1978
Co-author of the fundamental papers on Quantum Cryptography

10. Robert E. Bruccoleri, Ph.D.
Principal Scientist

Department of Macromolecular Modeling
Bristol-Myers Squibb Pharmaceutical Research Institute

Primary author of the program CONGEN, a mulitfunctional molecular modeling package, which is used to predict the structure of proteins by homology. He obtained his B.S. in Computer Science from the State University of New York at Stony Brook, his M.S. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, and his Ph.D. in Biochemistry and Molecular Biology from Harvard University.

11. Matthias Bruestle
Student of Chemistry at the University of Erlangen
PGP — Administrator of the KommunikationsNet Franken e.V.

12. Steve Byrne

Mr. Byrne is a graduate of Carnegie-Mellon University, and has been in the computer industry for 15 years. He has worked with everything from operating systems, to expert systems, to graphical user interface based end-user applications. He has developed software on everything from 8 bit microprocessors to mainframes, and has over 20 years of experience working with computers, and is regarded both as a senior design and architectural contributor, as well as an experienced user interface designer.

Mr. Byrne is the creator of the GNU version of the Smalltalk programming language. He has also created implementations of several other programming languages, as well as several utility programs and packages. His current interests are object oriented systems, distributed programming, programming languages, and graphical user interface design and implementation.

13. James Carlson

I am currently a Principal Software Engineer for Xylogics, Incorporated, and I develop communications software

for the Annex line of terminal and communications servers, which are used by many Internet dial-up sites and universities as the interface between the network and the telephone system.

14. Mike Coleman

Mike Coleman holds a Master's degree in Computer Science from UCLA and has a number of years of industrial software development experience, including programming language and user-interface design. He is presently at the Center for Telecomputing Research at the University of Missouri-Kansas City.

15. Richard H. Crawford

Currently a Researcher in Computer Security at the University of California, Davis. His three years of work experience with Hewlett-Packard as a Software Development Engineer involved designing user interfaces to database manipulation programs.

Crawford received a B.S. in Mechanical Engineering from UC Berkeley in 1979, followed by a M.S. in Computer Science from UC Davis in 1991. After completing his M.S. Thesis, "Topics in Behavioral Modeling and Event-Based Debugging", Crawford continued to pursue several strands of this research.

16. Pavel Curtis

Dr. Pavel Curtis has been a member of the research community at the Xerox Palo Alto Research Center since 1983, during which time he has worked on aspects of the Smalltalk-80, Interlisp-D/Xerox Lisp, and Cedar programming environments and on other projects mostly related to the design and implementation of programming languages, including leadership of the SchemeXerox project exploring large-scale software development in the Scheme programming language. He is the founder and chief administrator of LambdaMOO, one of the most popular recreational social virtual realities on the

Internet. His current work centers on the Network Places project, investigating the implementation, applications, and implications of systems that allow multiple simultaneous users to communicate and interact in pseudo-physical surroundings.

17. Josef Dalcolmo

I am a Senior Development Engineer (holding a Ph.D. in Physics) at the University of California at Berkeley. I have been working for the Austrian Academy of Sciences, the European Space Agency and for the last six years for the University of California at Berkeley, Space Sciences Laboratory.

18. Hugh Daschbach

Hugh Daschbach has been a professional programmer since 1973. He has worked on numerous projects for both computer users and vendors. Now employed as an independent software consultant, he has worked on projects for Digital Research, IBM, Xerox, FileNet Corporation, and the Aluminum Company of America.

19. L. Peter Deutsch

Dr. L. Peter Deutsch received the Ph.D. in Computer Science from U.C. Berkeley in 1973. Prior to the Ph.D. he was one of the key designers and implementors of the SDS 940 time-sharing system, the first commercial, general-purpose time-sharing system using paging hardware. In his subsequent 13 years at Xerox PARC, where he attained the position of Research Fellow, he was one of the key designers and implementors of the research prototype of the Interlisp-D system; a significant contributor to the design of the Cedar Mesa language and the Smalltalk-80 programming environment; and the principal creator of PS, the first high-performance implementation of the Smalltalk language and programming environment on microprocessor-based hardware. From 1986 to 1991, Dr. Deutsch was Chief Scientist at ParcPlace Systems, where he was a principal designer of a high-performance

Smalltalk implementation that is also highly portable across processors, operating systems, and window systems. From 1991 to early 1993, Dr. Deutsch held the position of Sun Fellow at Sun Microsystems, where he helped define future corporate strategy and technology in a variety of areas. In 1993, Dr. Deutsch founded, and from 1993 to 1995 served as President of, Artifex Software Inc., which commercially licenses a highly portable high-performance implementation of the PostScript language developed by Dr. Deutsch that is also available for non-commercial use from the Internet under the name Ghostscript. Dr. Deutsch was a co-recipient of the ACM Software System Award in 1993, and in 1993 was also named a Distinguished Alumnus of the U.C. Berkeley Computer Science program. Dr. Deutsch is a member of ACM, IEEE, CPSR, and the League for Programming Freedom.

20. Antonio J. De Vido

Systems Administrator, Sanwa Financial Products, L.P. B.S. Computer Science, Yale University 1993.

21. Robert B. K. Dewar

Professor of Computer Science and Associate Director of the Courant Institute of Mathematical Sciences at New York University. Active in operating systems and programming languages design for thirty years. Principle designer and author of the SPITBOL system. Chairman of IFIP WG2.1. Member of ISO standardization committees. Author of "Microprocessors: A Programmer's View", McGraw Hill 1990. Active in the design and development of the Ada language. One of the principle authors and designers of the GNAT system, a completely free software implementation of Ada 95 in current use at thousands of sites world wide. President of Ada Core Technologies, a company dedicated to Free Software and the encouragement of the use of Ada 95.

22. Mark-Jason Dominus

I was a systems programmer for the University of Pennsylvania Department of Computer and Information Sciences from 1990-1994, when I left to help start 'Pathfinder', Time-Warner's Internet information service. I now run Plover Systems, which provides consulting services to people who want to set up Internet-based information services.

23. Lester Earnest

Education:
B.S. Electrical Engineering, Caltech, 1953
M.S. EE/Computer Science, MIT, 1960
Professional experience:
1988-present: Consultant, Los Altos Hills, CA
1985-1988: Stanford University, Stanford, CA; Associate Chairman of Computer Science Department
1980-85: Founding President of Imagen Corporation, Santa Clara, CA, which introduced the first desktop publishing systems using laser printers.
1965-80: Executive Officer, Artificial Intelligence Laboratory, Stanford University, Stanford, CA.
1963-65: Department Head, Information Systems Dept., Mitre Corp., Washington, DC
1958-63: Subdepartment Head, Intelligence Systems Dept., Mitre Corp., Bedford, MA.
1956-57: Member, Technical Staff, MIT Lincoln Laboratory, Lexington, MA.

Invented the first computer spelling checker (1967). Created the first pen-based computer system that reliably recognized cursive handwriting (1961). Created the first FINGER program (1973), which has since become a standard utility on most computer systems on the Internet. Was a member of the technical committee that evaluated initial bidders on the creation of ARPAnet, which became the Internet. Organized and led the first successful robotics project at Stanford, which employed a TV camera and

mechanical arm (1966-68). Later initiated a visually-guided vehicle project (Stanford Cart, 1969), which led to a number of Ph.D. dissertations.

24. Paul Eggert
Director, R&D
Twin Sun Inc.
El Segundo, CA

Paul Eggert has directed software research and development projects at Twin Sun since 1989 in the areas of networked multimedia, computer supported cooperative work, and software reliability. From 1986 to 1989, Eggert was Senior Advisor to Management at Unisys Corporation, mostly working in the area of software verification. From 1983 to 1986, he was technical director and then Chief Scientist at Silogic Inc, directing development of logic programming software tools. From 1980 to 1983, he was Assistant Professor of Computer Science at the University of California, Santa Barbara.

25. Thomas G. W. Epperly

Computational Chemical Engineering. Education: BS Chemical Engineering, Carnegie-Mellon University, 1988; PhD Chemical Engineering, University of Wisconsin-Madison, 1995. Honors & Awards: Computational Science Graduate Fellowship (U.S. Dept. of Energy) 1991-95. Experience: Research Associate, Imperial College, England.

26. Michael Ernst

Lecturer at Rice University. He has performed research in the areas of artificial intelligence, parallel algorithms, compilers, cryptography, intellectual property, compilers, and static analysis. He has a Bachelor's and Master's degree from MIT.

27. Craig A. Finseth

Author of the main book on how to write text editors: "The Craft of Text Editing, or, Emacs for Modern Times."

Also, author of several programs distributed under GNU CopyLeft including Freyja (an Emacs for very small machines) and Loki (a calculator for the same). I am the manager of the data network for the State of Minnesota (~53,000 computers), an area in which it is crucial that users have access to multiple vendors that implement the same interfaces.

28. Simson L. Garfinkel
Author, Practical UNIX Security (O'Reilly & Associates, 1991)
PGP: Pretty Good Privacy (O'Reilly & Associates, 1994)
NeXTSTEP Application Programming (1993)

29. Daniel E. Geer, Jr., Sc.D.

Head of engineering for special projects at Open Market, Inc., a market leader in electronic commerce technology with corporate headquarters in Cambridge, Mass. Dr. Geer has been a successful entrepreneur in network security and distributed systems management, culminating in a successful sale of his own company to OpenVision Technologies, where he subsequently served as Chief Scientist, V.P. of Technology, and Managing Director of their security consulting services. Previously, he served as Technical Director, Innovation Technology Resource Center, Digital Equipment Corporation, and was Manager of Systems Development for MIT's Project Athena where he was the responsible manager for all technical development, including X, Kerberos, and all other aspects of the Project Athena Network Services System. He holds a Bachelor of Science in Electrical Engineering and Computer Science from MIT, and a Doctor of Science in Biostatistics from Harvard University, and was deeply involved in medical computing for fifteen years.

30. Howard Goldstein

Mr. Goldstein is President and CEO of InfoMotion Inc., a Florida software engineering firm that develops software

for the wireless data-communications industry. He graduated *cum laude* from Stetson U. College of Law, and is a member of The Florida Bar.

31. Allan Gottlieb
Professor of Computer Science, New York University
Director, NYU Ultracomputer Research Laboratory
Co-editor, Journal of Parallel and Distributed Computing

32. Paul Haahr
Employment: Harlequin Inc, senior software engineer, 1994-present.
Kaleida Labs, Inc., senior software engineer, 1993-94.
Adobe Systems Inc., computer scientist, 1990-93.
Polygen, Inc., 1987, systems administrator.
Oracle Systems, Inc., 1986-87, member of technical staff.
Education: Princeton University, AB in computer science, summa cum laude, 1990.
Published work: Contributor to Dylan Language and Reference Manual, Apple Computer, 1995. "Es: A shell with higher-order functions," co-authored with Byron Rakitzis, Winter 1993 Usenix conference proceedings. "Montage: Breaking windows into small pieces," Summer 1990 Usenix conference proceedings.

33. Jeffrey R. Hagen

Currently Senior Scientific Programmer Analyst with the National Radio Astronomy Observatory in Tucson Az.

Past five years responsible for the real-time control system at the Twelve Meter Radio Telescope on Kitt Peak. Wrote the 'RAMBO' interpretive C language which we use to define observing modes at the twelve meter. Wrote the telescope operators interface and well as others.

34. Darrel Hankerson
Associate Professor Department of Discrete and Statistical Sciences

120 Mathematics Annex
Auburn University

35. Diane Harrison
Business programmer at Triad since 1979

36. Brian Harvey
Lecturer, Computer Science Division, University of California at Berkeley

I am the author of several books about computer science, and the chief implementor of Berkeley Logo, a freeware version of the Logo programming language.

37. Jeff Hay

Computer Science undergrad at New Mexico Tech. Has several small programs in the public domain/shareware markets.

38. Christopher T. Haynes

Christopher T. Haynes is Associate Chair for Education and Associate Professor in the Computer Science Department of Indiana University at Bloomington, where he directs the Scheme Educational Infrastructure project funded by the National Science Foundation. He chaired the IEEE Scheme Working Group that developed the IEEE and ASCII Scheme standard and has been from its inception a member of the Scheme report author's group. Professor Haynes is co-author of the MIT Press/McGraw Hill text "Essentials of Programming Languages", which uses Scheme throughout. Professor Haynes has authored over a dozen programming language research papers. As well as teaching at Indiana University since 1982, he has taught graduate courses at MIU in Fairfield, Iowa, IISc in Bangalore, India, and ITESM in Monterrey, Mexico, and has consulted with industry. He received a PhD degree in computer science from the University of Iowa in 1982 and a BA degree in mathematics from Lehigh University in 1973.

39. Richard F. Hilliard, II

Lead systems architect, at the MITRE Corporation. Active in ISO, ANSI and U.S. Department of Defense Standardization activities. Member of the Ada 9X Mapping/Revision Team.

40. Michael D. Hirsch
Assistant Professor of Mathematics and Computer Science
Emory University
AB: Princeton 1984; PhD: Berkeley 1990
NSF Postdoc: 1990-1993

41. Christian D. Hofstader

Chris Hofstader is a co-founder, director and the primary organizer of the League for Programming Freedom. He is also a programmer and has contributed to many popular PC products including: The Bank Street Writer, Portfolio's Dynodex and WordPerfect Works for Windows. His recent projects include the design of user interfaces for blind and low vision users of personal computers.

42. Max Hyre
Staff Engineer
General Signal Networks
Shelton, Connecticut 06484

I have been a programmer of embedded communications systems for 17 years, and as such my work has relied critically on the availability of multiple programs presenting the same user interface (such as text editors & word processors, compilers, and debugging programs).

43. Philip Jensen

I have been working in computer software since high school days. I hold the degree of Bachelor of Arts in Applied Mathematics, granted by Harvard College. I am currently employed by CompuServe, Inc.

44. Jim Johnson

I have been a published computer industry columnist since June of 1990, covering career issues and industry trends. I'm also a Certified Personnel Consultant (CPC), and have served as a personnel consultant to the computer industry since 1987.

45. Phil Karn

I am an engineer in the communications industry with 16 years of professional experience. I have written a computer networking software package that is very widely used to provide low-cost access to the Internet over ordinary telephone lines. I am currently employed in the digital wireless communications industry and reside in southern California.

46. John Kohl
Systems Software Programmer, 10 years experience.

Mr. Kohl has authored technical papers on computer security and file system storage, and many freely-available software packages including hardware device driver modules, disk filing system modules, and network authentication programs. He holds a MS in computer science from the University of California, Berkeley, and a BS in computer science from the Massachusetts Institute of Technology.

47. Markus Krummenacker
Software Consultant

1994-1995: Director of Research, Nanothinc, San Francisco, California. Key work in overall layout of WWW server and company strategy. 1992-1993: Research Associate, Institute for Molecular Manufacturing, Palo Alto, California. Analysis of Molecular Building Blocks, and writing the CavityStuffer Program (in CommonLISP). 1990-1992: Visiting Researcher, Dept. of Molecular and Cell Biology, University of California at Berkeley. Isolation via PCR, and sequence determination of homologous genes for the Ferric Uptake Repressor in bacteria.

48. Andrew Kuchling

Andrew Kuchling is a native of Canada, currently working as a programmer for a Montreal software firm that is a leader in the field of corporate calendaring software. His interests include cryptography, novel programming languages, and general network applications, such as the World-Wide Web. Among other things, he maintains the major toolkit of cryptography extensions for the Python programming language.

49. Phil Laird

Dr. Philip Laird is a Computer Scientist with the Computational Sciences Division at NASA's Ames Research Center in Mountain View, California.

50. Charles Landau
Tandem Computers
Cupertino, CA 95014
BSEE, MIT, '71; MSEE, MIT, '72
Founder, Key Logic Inc. (a software company)
Founder, MACS Lab, Inc.
Architect of the KeyKOS operating system.
Author of several published papers on KeyKOS and other subjects.
Computer programmer from 1968 to the present.

51. Emery W. Lapinski

I hold a BA in CS from NYU and have been professionally designing and implementing GUIs for applications used in the Banking and Telecommunications industries for the past few years.

52. James E. Leinweber

I have used computers for over 23 years, and have been a full-time professional computer programmer, analyst and administrator since 1984. I currently work for the Wisconsin State Laboratory of Hygiene, a public health

Lab at the University of Wisconsin in Madison. I hold a Master's degree in Mathematics from Michigan State University, and am a member of Phi Beta Kappa, Computer Professionals for Social Responsibility, and the League for Programming Freedom.

53. John McCarthy

John McCarthy is Professor of Computer Science at Stanford University. He has been interested in artificial intelligence since 1948 and coined the term in 1955. His main artificial intelligence research area has been the formalization of common sense knowledge. He inv nted the LISP programming language in 1958, developed the concept of time-sharing in the late fifties and early sixties, and has worked on proving that computer programs meet their specifications since the early sixties. He invented the circumscription method of non-monotonic reasoning in 1978.

His main research (1995) is formalizing common sense knowledge and reasoning. His most recent paper is on formalization of context.

McCarthy received the A. M. Turing award of the Association for Computing Machinery in 1971 and was elected President of the American Association for Artificial Intelligence for 1983-84 and is a Fellow of that organization. He received the first Research Excellence Award of the International Joint Conference on Artificial Intelligence in 1985, the Kyoto Prize of the Inamori Foundation in November 1988, and the National Medal of Science in 1990. He is a member of the American Academy of Arts and Sciences, the National Academy of Engineering and the National Academy of Sciences. He has received honorary degrees from Link\"oping University in Sweden, the Polytechnic University of Madrid, Colby College, Trinity College, Dublin and Concordia University in Montreal. He has been declared a Distinguished Alumnus by the California Institute of Technology.

54. Chris McCraw

Network administrator for the Texas Institute of Computational & Applied Mathematics; contract work-Internet/heterogeneous networking/Internet + Unix focus.

55. Steven W. McDougall
Education:
BSEE, Princeton University 1982; MSE, Stanford University 1983
Employment:
1983-1987 Electrical Engineer, General Computer Co. Waltham MA; 1987-1990 Electrical Engineer, OsteoTechnology, Inc Framingham MA; 1990-1992 Software Engineer, Varityper, Billerica MA; 1992-1993 Software Engineer, Sepracor, Marlbourough MA; 1993-1995 Software Engineer, Genome Therapeutics,Waltham MA

56. Geoff Mendal
Vice President of Engineering
Tri-Pacific Consulting Corporation

Technical lead of a large-scale porting effort for Sun Microsystems, Inc.: the SunSoft Workshop (C, C++, ProWorks, and TeamWare tools) is being ported from SunOS 5.x to HP-UX 9.0x. Co-author of 2 research/reference texts on the programming language Ada (Exploring Ada, volumes 1 and 2).

57. Emilio Millan

Software engineer for Central Data Corporation of Champaign, Illinois.

58. Clark Morgan

Clark O. Morgan is an independent consultant and full-time computer software engineer, currently working on a contract basis with Intel Corp. I have been working full-time in the computer industry since 1984. My particular

areas of interest are language recognition and the development of embedded, real-time hardware/software interfaces. My accomplishments include the development of an instrumentation control language for a digital oscilloscope (at Tektronix, Inc.), the development and maintenance of a Motif-compatible e-mail browser (as an employee of Mentor Graphics, Corp.), and the development of a paperless "run card" for Intel's premier wafer fabrication research facility in Aloha, Oregon.

59. Neil Vincent Murray
Associate Professor, Department of Computer Science, SUNY at Albany, Albany, New York 12222, 1987-present.
Assistant Professor, Department of Computer Science, SUNY at Albany, Albany, New York 12222, 1982-1987.
Assistant Professor, Computer Science Department, LeMoyne College, Syracuse, New York 13214, 1979-1982.
Cornell University, Ithaca, New York 1966-1970, B.S. Engineering Physics, June 1970
Syracuse University, Syracuse, New York 1971-1979
M.S. Computer and Information Science, January 1974,
Ph.D. Computer and Information Science, August 1979

60. Russell Nelson

Russell Nelson has been programming professionally since 1975. He has written at least one of nearly every type of program, including compilers, interpreters, macro-based text editors, and other types of programs related to this brief. His work has been recognized by PC Magazine for Technical Excellence in 1990. Somewhere between a hundred thousand and a million people are using his software.

61. Jeffrey F. Painter
B.S., Mathematics, Harvey Mudd College, 1973
M.S., Computer Science, University of Wisconsin-Madison, 1977

Ph.D., Applied Mathematics University of Wisconsin-Madison, 1979
Mathematician, Lawrence Livermore National Laboratory (University of California), 1979-present. I write computer codes for scientific simulations. I also developed a user interface and input language for a program that partly automates the development of scientific simulation codes. Articles I wrote are in journals and books on mathematics, engineering, and computer science.

62. Kent M. Pitman
Principal Technical Consultant, Harlequin Inc. [*]
Project Editor [*] and International Representative [*] for subcommittee X3J13 (ANSI programming language Common Lisp, X3.226). US Representative [*] to and Project Editor [*] for subcommittee ISO/IEC JTC 1/SC 22/WG 16 (work in progress toward design of ISO programming language ISLISP).
Co-Author of "Revised Report on the Algorithmic Language Scheme", MIT AI Memo 848b, Cambridge, MA, Jul-Sep 1991.
* = Title indicated shows affiliation but the signature offered here is not offered as official act of representation under that title. The title is offered here for purpose of identification and establishment of credentials only.

63. Steven Pothier

Mr. Pothier is a Senior Computer Scientist for the SAIC Image Technology and Systems Division. He works on state-of-the-art software systems incorporating object-oriented languages, advanced decision techniques, reconfigurable user interfaces, and genetic programs. Mr. Pothier is an expert-level C, and LISP programmer and has considerable experience in many other high and low-level languages many of which are high-level (4GL) languages. He has several years experience programming SIMD and MIMD parallel processors, including experience with data-parallel languages for the Connection Machine Model 2 (*Lisp, C/Paris, C*).

64. John Ramsdell

John Ramsdell is a Lead Scientist at The MITRE Corporation, Bedford MA. He has been employed there for 12 years. In 1982, he received his Ph.D. in Applied Mathematics, concentrating in numerical and discrete methods, from Harvard University. In 1976, he received his B.S., in Engineering Physics, from Cornell University.

At MITRE, John Ramsdell specializes in parallel and functional programming, high performance computation and communication, and formal verification. His activities have contributed to the construction of a verified processor which executes functional programs, and the construction of a formally verified implementation of a programming language (Scheme). He is involved in an effort to replace special purpose hardware for signal processing by commercial massively parallel processors.

65. Lyle Ramshaw

After getting my PhD under Prof. Donald E. Knuth at Stanford in 1979, I have been doing computer science research, first for Xerox Corporation and now for Digital Equipment Corporation. I pioneered "blossoming", a new approach to the mathematical theory underlying the "spline" curves and surfaces that are used in computer-aided geometric design.

66. Travis Raybold

I am a computer programmer, dealing mainly in cgi scripting for the world wide web.

67. Mark Riordan

Mark Riordan is the original author of the popular freeware Usenet newsreader WinVN, and of a free encryption program named RIPEM.

68. Arnold Robbins

I am the maintainer of GNU AWK, a freely available implementation of the AWK computer programming language. My involvement with this program has greatly improved my programming and writing skills, and has allowed me to improve the lot of hundreds of thousands of people who use GNU AWK daily for their work.

69. Michael C. Robert
Project Leader

I have been programming user interfaces for 11 years and I am currently working for a consulting company that designs and builds mass-market applications for Fortune 500 companies. To be an effective programmer, I have had to also become quite familiar with users and their expectations and needs.

70. Erik Rosenthal

I am a professor in the department of mathematics at the University of New Haven; my research interests are in the area of automated deduction. My Ph.D. is from Univ. of Calif. at Berkeley.

71. Roger Schlafly

Independent software developer. Ph.D. in Mathematics, Univ. of California (Berkeley). Worked in the software industry for ten years.

72. Karl F. Sierka
Boulder, Colorado

I am currently working for an ISP, doing usenet news admin. I have been doing computer work since the early eighties. I have written several large programs in several languages, each requiring extensive interactions with users.

73. Rafael Sorkin

He has degrees in physics from Harvard (undergrad) and Caltech (grad). Prof. of Physics at Syracuse University, on editorial board of International J. of Theoretical Physics. One of first people to use computers for numerical General Relativity (in around 1973).

74. Todd Squires

Graduate of University of Illinois at Urbana, 1987, B.S. Computer Science. Working in the computer industry for 15 years. Major areas of concentration: Video games, development tools.

75. Gerald Jay Sussman

Gerald Jay Sussman is the Matsushita Professor of Electrical Engineering at the Massachusetts Institute of Technology. He received the S.B. and the Ph.D. degrees in mathematics from the Massachusetts Institute of Technology in 1968 and 1973, respectively. He has been involved in artificial intelligence research at M.I.T. since 1964. He has also worked in computer languages and in computer architecture and in VLSI design.

Sussman was the principal designer of the Digital Orrery, a machine designed to do high-precision integrations for orbital-mechanics experiments. Using the Digital Orrery, Sussman and Jack Wisdom, discovered numerical evidence for chaotic motions in the solar system. The Digital Orrery is now retired at the Smithsonian Institution in Washington DC.

Sussman is a coauthor (with Hal Abelson and Julie Sussman) of, "Structure and Interpretation of Computer Programs," the introductory computer science textbook used at M.I.T. As a result of his contributions to computer-science education, Sussman received the ACM's Karl Karlstrom Outstanding Educator Award in 1990, and the Amar G. Bose award for teaching in 1991.

76. Dr. Christopher Vickery

Dr. Christopher Vickery is professor of computer science at Queens College of CUNY. He is the author of the volume, "Real-Time and Systems Programming for PCs" (McGraw-Hill, 1993), and is listed in the current edition of Marquis' Who's Who in the East. Dr. Vickery has developed a number of software development tools for

microprogrammable computers and real-time systems, which have been distributed freely to coworkers in the industry.

77. Philip Wadler

Philip Wadler is a Professor of Computing Science at the University of Glasgow. He serves as editor-in-chief of the Journal of Functional Programming, published by Cambridge University Press; and wrote with Richard Bird 'Introduction to Functional Programming', which has been translated into Dutch, German, and Japanese. He helped design Haskell, the most widely used lazy functional programming language; and currently consults with Ericsson Telecommunications on the design of a type system for Erlang, a concurrent functional language used in phone switches. He has been an invited speaker at conferences in Gdansk, London, New Haven, Portland, Santa Fe, and Sydney; and was an ACM distinguished lecturer 1989-1993. He previously held a research post at Oxford University, and the title of his doctoral dissertation from Carnegie-Mellon was 'Listlessness is better than laziness'.

78. John Wardale
BSECE 1983, MSCS 1985 [UW-Madison]
Jobs: Engineering Computing Lab
Computer Aided Engineering Center
Astronautics Technology Center [unix-OS group member for the ZS-1 mini-super computer group]
Persoft Inc. [R&D team member—application programer]

79. Joel Welling

Joel Welling holds a PhD in gravitational physics. For almost a decade he has worked at the Pittsburgh Supercomputing Center, where he writes software for scientific computer graphics in support of the research of academic scientists in a number of locations across the US. He has written several computer graphics packages (including some supporting the CGM standard graphics

metafile, the P3D system for three dimensional graphics, and the VFleet volume rendering package) which are available free via the Internet.

80. Ben White

Ben White is a computer scientist, with an M.S. degree from San Jose State University. He has 11 years of experience developing both computer software and hardware, including development of a visual programming environment called "Flowtran". He is currently employed with a maker of semiconductor manufacturing equipment. He has been a member of the League for Programming Freedom for four years.

81. Jan Zytkow

Jan Zytkow received his Ph.D. from the University of Warsaw in Philosophy of Science in 1972, and his habilitation in 1979. Until 1982 he worked at the University of Warsaw, researching and teaching logic, methodology of science, and philosophy. From 1982, for two years he was a Visiting Professor at Carnegie-Mellon University in Pittsburgh, where he worked on machine discovery, collaborating with Herbert Simon and Pat Langley. In 1984 he joined the Wichita State University where he has been a Professor of Computer Science and in addition to his research on discovery, he worked on autonomous decision making by automated pilot, on learning mechanisms for automated pilot, and on computer vision. He founded a machine discovery laboratory in which automated computer systems together with robotic hardware perform experiments and build theories in chemistry, mechanics and the like. Other research activities in the laboratory include discovery of hidden structure, discovery in databases, and discovery in geometry.

MOTION FILED
DEC 7 1995

No. 94-2003

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORP.,
Petitioner

vs.

BORLAND INTERNATIONAL INC.,
Respondent

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

Motion For Leave To File Brief Amicus Curiae and Brief Amicus Curiae of Howard C. Anawalt In Support of Respondent

CAROL A. KUNZE
50 Palm Ave.
San Francisco, California
94118
(415) 386-1785

HOWARD C. ANAWALT
Counsel of Record
University of Santa Clara
Santa Clara, California
95053
(408) 554-4090

Counsel for Amicus Curiae

BOWNE OF PALO ALTO • 1800 EMBARCADERO RD. • PALO ALTO, CA 94303 • (415) 856-1800

BEST AVAILABLE COPY

18 pp

Pursuant to Rule 37.3(b) of the Supreme Court of the United States, I respectfully move for leave to file the accompanying brief *amicus curiae* which argues for affirmance of the First Circuit Court of Appeals decision in this case. Consent of the attorneys for Borland International, Inc. was requested by letter dated October 25, 1995, and by telefax and letter dated November 20, 1995, but has been refused by telephone on November 28, 1995. Consent of the attorneys for Lotus Development Corporation has been requested by letter dated October 25, 1995, by telefax and letter sent by certified mail on November 20, 1995, and through several telephone messages, but has not been received.

As a professor of law who teaches courses in the protection of intellectual property with a particular emphasis on high technology, I am vitally interested in the development of the law of copyright as it pertains to the protection of computer programs. I believe there is tension in the law as applied to computer programs because of their functional nature, and that the statutory restriction on copyrights which bars protection for ideas, processes, *etc.*, Section 102(b) of the Copyright Act, 17 U.S.C. §102(b), mandates affirmance of the Court of Appeals decision.

The analysis in this brief presents the view that Section 102(b) refers to an integrated domain of matter, including ideas and utilitarian or functional applications of those ideas, rather than a list of discrete items barred from copyright protection. I do not believe this particular view is otherwise presented in the briefs submitted in this case. I conclude that the menu command hierarchy of Lotus 1-2-3 constitutes a function which falls within the latter portion of this integrated domain, and it must therefore be denied copyright protection.

For the reasons set forth above, I respectfully request that this Court grant this motion for leave to file the accompany-

ing brief *amicus curiae* arguing for affirmance of the First Circuit Court of Appeals decision.

Respectfully submitted,

CAROL A. KUNZE
50 Palm Ave.
San Francisco, California
94118
(415) 386-1785

HOWARD C. ANAWALT
Counsel of Record
University of Santa Clara
Santa Clara, California
95053
(408) 554-4090

Counsel for Amicus Curiae

Dated: November 29, 1995

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	iv
INTEREST OF *AMICUS CURIAE*	1
SUMMARY OF ARGUMENT	1
ARGUMENT	2
INTRODUCTION	2
I	
Section 102(b) excludes ideas and predominantly utilitarian elements from copyright protection	2
A. Section 102(b) excludes ideas from copyright protection	5
B. Section 102(b) excludes utilitarian elements from copyright protection	6
II	
Copyright does not extend to those aspects of a computer interface which constitute the command function	8
III	
Conclusion	12

TABLE OF AUTHORITIES

Cases

	Page
Affiliated Hosp. Prod., Inc. v. Merdel Game Mfg. Co., 513 F.2d 1183 (2d Cir. 1975)	10
Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984)	5
Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992)	7
Baker v. Selden, 101 U.S. 99 (1879)	6
Cohen v. California, 403 U.S. 15 (1973)	11
Computer Assoc. Int'l, Inc. v. Altai , Inc., 775 F.Supp. 544 (E.D.N.Y. 1991), *aff'd in part*, 982 F.2d 693 (2nd Cir. 1992)	10
Feist Publications, Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991)	5
Gottschalk v. Benson, 409 U.S. 63 (1972)	7
Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F.Supp. 203 (D. Mass. 1992)	9
Lotus Dev .Corp. v. Borland Int'i, Inc., 49 F.3d 807 (1st Cir. 1995)	2, 8, 9
Mazer v. Stein, 347 U.S. 201 (1954)	8
Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)	5
Olson v. Nat'l. Broadcasting Co., 855 F.2d 1446 (9th Cir. 1988)	3
R.A.V. v. City of St. Paul, 505 U.S. 377 (1992)	11

TABLE OF AUTHORITIES

CASES

	Page
Sega Enter., Ltd. v. Accolade, Inc., 977 F. 2d 1510 (9th Cir. 1992)	10
Texas & New Orleans R.R. Co. v. Brotherhood of Ry. & S.S. Clerks, 281 U.S. 548 (1930)	3
Third Nat'l Bank in Nashville v. Impac Ltd., 432 U.S. 312 (1977)	3

Statutes

17 U.S.C.	§101	7
	§102(b)	*passim*
35 U.S.C.	§100(b)	6
	§101	6, 7
	§101(a)	6
	§102(a)(8)	7
	§102(g)	6
	§112	6
	§171	7
U.C.C. §3-110(1)		11

Legislative Materials

H.R. Rept. No. 1476, 94th Cong., 2d Sess. (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659	4, 8

Other Authorities

The American Heritage College Dictionary (3rd ed. 1993)	3, 6, 9
Marc McCutcheon, *Roget's Superthesaurus* (Writer's Digest Books 1995)	2, 3

Interest of *Amicus Curiae*

I am a professor of law and director of the High Tech Law Program at the School of Law, University of Santa Clara, Santa Clara, California. I teach courses on the protection of intellectual property and on computer law, and am the author of the book *Ideas in the Workplace: Planning for Protection* (1988) and co-author of *Licensing Law Handbook* (1991). I have no financial interest in either party, nor in the outcome of this case. My interest is based on the impact this case will have on the current law regarding the application of copyright protection to computer programs. I believe that affirmation of the First Circuit Court of Appeals decision is necessary to uphold the statutory limitation which confines copyright to expressive non-functional works.

SUMMARY OF ARGUMENT

Section 102(b) of the Copyright Act, 17 U.S.C. §102(b), refers to an integrated domain of matter, including ideas and utilitarian or functional applications of those ideas, rather than a list of discrete items barred from copyright protection. The menu command hierarchy of Lotus 1-2-3 constitutes a function which falls within the latter portion of this integrated domain, and it must therefore be denied copyright protection.

ARGUMENT

Introduction

This friend of the court brief is submitted to support the decision of the First Circuit Court of Appeals which correctly excluded a computer program command hierarchy from copyright protection pursuant to Section 102(b) of the Copyright Act, 17 U.S.C. §102(b).

The menu command hierarchy in this case represents the set and sequence of commands necessary to make the particular program perform tasks that its human operator has set out for it.[1]

I

Section 102(b) excludes ideas and predominantly utilitarian elements from copyright protection

Section 102(b) states its exclusion from copyright in clear and direct terms: copyright protection does not extend to ideas, procedures, processes, systems, methods of operation, concepts, principles, or discoveries.

Section 102(b) denotes two related types of matter, combined into an integrated realm which is excluded from copyright, rather than a listing of discrete topics of non-copyrightable subject matter. "Procedure," "process," "system," and "method" are synonyms or convey similar meanings;[2] "discovery" conveys the idea of obtaining "knowledge

[1]". . . Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy." *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 810 (1st Cir. 1995).

[2]Synonyms for "process" include "method" and "procedure"; for "method" include "procedure," "process," "system." Marc McCutcheon, *Roget's Superthesaurus* 403 (Writer's Digest Books 1995) [hereinafter *Roget's*].

of, as through study."[3] All four words are also used to mean patentable subject matter.[4] While dictionaries provide different specific meanings to "idea," "concept" and "principle," all three words contain a deep common thread in that they denote thoughts at a high level of abstraction or generality.[5]

Section 102(b) is an instance where the words support each other. "It is a familiar principle of statutory construction that words grouped in a list should be given related meaning."[6]

The legislative history of Section 102 confirms that Congress intended to exclude a general sphere of matters. In explaining the section, the House Report stated:

> Copyright does not preclude others from using the ideas or information revealed by the author's work. It pertains to the literary, musical, graphic, or artistic *form in which the author expressed intellectual concepts*. . . .
>
> Some concern has been expressed lest copyright in computer programs should extend protection to the

[3]The American Heritage College Dictionary 396 (3rd ed. 1993) [hereinafter *Dictionary*].

[4]See *infra* text accompanying notes 11 and 12.

[5]Indeed, *Roget's* lists "concept" as a synonym for "idea." *Roget's, supra* note 2, at 261. The word "concept" may denote or connote a certain tinge or variety of idea: "a general idea derived or inferred from specific instances or occurrences." *Dictionary, supra* note 3, at 2ℓ8. A Ninth Circuit case notes "concept" and "idea" are synonyms in Section 102(b), *Olson v. Nat'l Broadcasting Co., Inc.*, 855 F.2d 1446, 1451 (9th Cir. 1988).

[6]*Third Nat'l. Bank in Nashville v. Impac Ltd.*, 432 U.S. 312, 322 (1977). The principle is also know by its Latin maxim, *noscitur a sociis*. *See also Texas & New Orleans R.R. Co. v. Brotherhood of Ry. & S.S. Clerks*, 281 U.S. 548, 568 (1930).

> methodology or processes adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression *adopted by the programmer is the copyrightable element in a computer program,* and that the actual processes or methods embodied in the program are not within the scope of the copyright law.
>
> Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. *Its purpose is to restate,* in the context of the new single Federal system of copyright, *that the basic dichotomy between expression and idea remains unchanged.*[7]

The quoted paragraphs point to a general concept that separates protectable from unprotectable matters. Protectable matter is the form in which one has "expressed intellectual concepts." On the other hand, ideas and strictly utilitarian applications of those ideas must remain outside the realm of protection.

The report also emphasizes that while computer programs may obtain copyright protection, they must remain subject to the same line of exclusion as other works. Congress addressed the concern that computer program copyrights would extend protection to the programmer's methodology or processes by reaffirming the rejection of protecting ideas in themselves. Because a computer program functions, in addition to representing expression, it presents more of a risk that its utilitarian ideas, its "methodologies or processes,"

---

[7]H.R. Rept. No. 1476, 94th Cong., 2d Sess. 56-57 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670 [hereinafter *House Report*]. Italics have been added.

might gain protection.[8] Section 102(b) assures that this will not happen.

A. Section 102(b) excludes ideas from copyright protection

The primary concept of Section 102(b) is that broad ideas, concepts and principles can not gain protection. This concept is firmly established in the famous *Nichols* case.[9] Ideas in themselves may not be copyrighted, because they appropriate far more than a particular expression. The ideas "love" and "freedom" are amongst the most expressive of human concepts. One can express these ideas in a huge number of different ways, yet the core ideas themselves cannot be held by copyright. Similarly, the general idea of organizing commands in a graphical menu is not protectable.

---

[8]The species of copyright protection that has evolved since *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984), represents a kind of "technological copyright" in that the expression protected actually becomes implemented in governing the physical processes of a machine. In computer programming the word "function" has a special meaning. The sentence in the body could also be phrased: "Because a computer program *directs how a computer operates*, in addition to representing expression, it presents more of a risk that its utilitarian ideas . . ."

[9]*Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930) (" . . . there is a point . . . where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended."). *See also Feist Publications, Inc. v. Rural Telephone Serv. Co.*, 499 U.S. 340, 344-5 (1991), stating: "The most fundamental axiom of copyright law is that '[n]o author may copyright his ideas or the facts he narrates.' ") (citations omitted).

B. Section 102(b) excludes utilitarian elements from copyright protection

A second aspect of Section 102(b) *combines with* the norm excluding ideas. This second strand excludes strictly utilitarian or functioning applications of ideas that are properly the subject of patent law.[10] Four of the words used in Section 102(b) specifically relate to matters that may be the subject of patents. Sections 100(b) and 101 of the Patent Act, 35 U.S.C. §§100(b) & 101, specifically include "processes" and "methods." Section 101(a) of the Patent Act also uses "discovery" as a synonym for "invention."[11] Furthermore, a "concept" once reduced to practice and applied in a concrete way constitutes patentable subject matter.[12] As stated by the Federal Circuit:

> Thus, patent and copyright laws protect distinct aspects of a computer program. *See Baker v. Selden,* 101 U.S. 99, 103, 25 L. Ed. 841 (1879). Title 35 protects the process or method; Title 17 protects the expression of that process or method.[13]

The legislative history directly affirms the distinction between the domain of patent law and that of copyright.[14] Patent covers utility.[15] Copyright covers non-utilitarian or non-functional expressive matters, with the notable exception of the "technological copyrights" presented by computer programs.[16]

The Copyright Act rejects protection of pure utility or function most clearly with regard to pictorial, graphic, and sculptural works. These can receive protection under the Act, but only to the extent that one excludes "their mechanical or utilitarian aspects."[17] The design of a "useful article," defined as an article "having an intrinsic utilitarian function," can be considered a protectable "pictorial, graphic, or sculptural work" only when its pictorial, etc. features "can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the work."[18]

---

[10] The utilitarian exclusion ties closely to the norm excluding ideas. If one separates the "utility" strand from the "abstraction/idea" strand in the analysis, the line of demarcation ought not be too sharply drawn. Section 102(b) excludes protection of abstraction and, especially, those matters that are utilitarian abstractions. A definition of method denotes a general means of doing something: "A means or manner of procedure, esp. a regular and systematic way of accomplishing something." *Dictionary, supra* note 3, at 857.

[11] "The term "invention" means invention or discovery." 35 U.S.C. §101(a).

[12] The concept must be reduced to practice so that when described in the patent specification it will "enable any person skilled in the art" to practice it. 35 U.S.C. §112. Priority in patent claims goes to the one who pursues the process from conception to reduction to practice with due diligence. 35 U.S.C. §102(g).

[13] *Atari Games Corp. v. Nintendo of Am., Inc.,* 975 F.2d 832, 839 (Fed. Cir. 1992). A patent application which fails to reduce a concept to practice runs afoul of the principle that an idea alone may not be patented, *see Gottschalk v. Benson,* 409 U.S. 63, 67 (1972). Neither copyright nor patent protects ideas in the abstract.

[14] *See supra* text accompanying note 7.

[15] With the exception of design patents. 35 U.S.C. §171.

[16] *See supra* note 8. "Architectural works" constitute a special group of particularly utilitarian works protected by copyright, but that class does not appear relevant to the present case. 17 U.S.C. §§101 (definition of "architectural work") & 102(a)(8).

[17] 17 U.S.C. §101 (definition of "pictorial, graphic, and sculptural works").

[18] 17 U.S.C. §101 (definition of "useful article").

The *House Report* indicates that protection was assured for pictorial, graphic, and sculptural works embodied in useful articles because Congress wished to continue protection for works of "applied art" which had previously been accorded by the decision of the Supreme Court in *Mazer v. Stein*, 347 U.S. 201 (1954). The report stated that in adopting the definition of "useful article" it sought "to draw as clear a line as possible between copyrightable works of applied art and uncopyrightable works of industrial design."[19]

Under the foregoing analysis, a menu command hierarchy constitutes a strictly functional or utilitarian element that must be denied protection under Section 102(b).

II

Copyright does not extend to those aspects of a computer interface which constitute the command function

According to both the District Court and the Court of Appeals, "the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms. . . ."[20] To illustrate this fact the District Court stated: "The 'Copy' command could be called 'Clone,' 'Ditto,' 'Duplicate,' 'Imitate,' 'Mimic,' 'Replicate,' and 'Reproduce,'

---

[19] *House Report, supra* note 7, at 55, 1976 U.S.C.C.A.N. at 5668.

[20] The Court of Appeals noted: "Under the district court's reasoning, Lotus's decision to employ hierarchically arranged command terms to operate its program could not foreclose its competitors from also employing hierarchically arranged command terms to operate their programs, but it did foreclose them from employing the specific command terms and arrangement that Lotus had used." 49 F.3d at 816. The Court of Appeals accepted "the district court's finding that Lotus developers made some expressive choices in choosing and arranging" the command terms. *Id.*

among others . . . ."[21] The Court of Appeals ruled that nonetheless these particular choices became part of an entire structure which constituted a "method of operation."[22] This conclusion is correct in that it has determined that the hierarchy falls within Section 102(b). The command hierarchy's sole purpose is to channel the precise human steps necessary to get the program to do its job. In order to use a computer program, a user must provide the commands that trigger operations controlled by the computer program itself.[23]

The Lotus hierarchy, including the names of individual commands, constitutes a *pure command function.*[24] In each case the key sequences input within the framework of the menu command hierarchy activate some response in the computer causing it to perform an operation. The menu command hierarchy, including its specific commands, is merely the tool used to cause the program to operate. Once

---

[21] *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203, 217 (D. Mass. 1992). Expressive identification of commands might well be protected. For example, "quit" might be expressed by "watermelon," "Kamo" (duck — Japanese), or an image of one punching out at a time clock. It appears that no such originality of expression is present in this case.

[22] 49 F.3rd at 815. The portion of Borland's Quattro Pro program at issue is the "Key Reader," which allows the program to interpret and perform some Lotus 1-2-3 macros. *Id.* at 811. In order to do so, the Key Reader file contains a copy of the Lotus 1-2-3 menu command hierarchy. *Id.* at 812.

[23] The Court of Appeals likened the commands to buttons used to control a VCR. *Id.* at 817. The authors of this brief have not examined the record, but rely on the construction of facts set forth in the published opinions for all factual observations.

[24] The definition of "command" comports with this observation, but the argument is not a semantic one. Command means "to direct with authority" or in computer science, "a signal that initiates an operation defined by an instruction." *Dictionary, supra* note 3, at 279.

utilized by a user to construct a macro (*e.g.*, a series of operations which can be executed through a single keystroke), its use is critical to the operation of the macro. The command hierarchy, including the command terms, thus constitutes pure function. As such, the hierarchy must be excluded from copyright by Section 102(b).

Examples from copyright and other branches of law that further recognize that words sometimes express ideas, while at other times they constitute action or serve a function, support the division between words as expression of ideas and the functional use of words. When this is the case, the law treats the words as something other than mere expression.

Rules of a game have been held to be not protectable by copyright.[25] The particular way in which the rules are described or expressed will be narrowly protected, however.[26] The menu command hierarchy presents a similar case. Expression, if any, found within the command hierarchy should only be protected to the extent of expressive value independent of the command function.[27]

---

[25] *Affiliated Hosp. Prod., Inc. v. Merdel Game Mfg. Co.*, 513 F.2d 1183, 1188-89 (2d Cir. 1975).

[26] *Id.*

[27] Intellectual property law has wrestled with the function/expression demarcation in a variety of other areas. We advert to some in this footnote. In *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) the Court rejected a trademark claim where the display of the mark "S-E-G-A" became a functional aspect of operation of the game itself, stating, that a "trademark owner may not enjoy a monopoly over the functional use of the mark." *Id.* at 1531. The trial court in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 775 F.Supp. 544, 559-560 (E.D.N.Y. 1991), *aff'd in part*, 982 F.2d 693 (2nd Cir. 1992), emphasized the behavioral aspect of a computer program in distinguishing between the static structure, sequence and organization of the computer program, represented by the program's copyrighted text, and the dynamic structure represented by the program's behavior. The court noted there was no relationship between the structure of the text (the source and object codes) and the sequence of operations in a program, which are behavior. Finally, the court stated the program's behavioral aspect fell within the statutory terms of "process," "system" and "method of operation" in Section 102(b). The static versus dynamic structure distinction was referred to approvingly by the Second Circuit in affirming the case. *Id.* at 706.

In other areas, when function overtakes expression, the law recognizes that the "expressive" or communicative aspect must be treated differently than when pure expression is involved. For instance, the words "pay to the order of" endorsed on the back of a check function to transfer the rights in the check to another.[28] In common experience one acknowledges when function overtakes expression by responding with action rather than words. Giving the military command of "fire" to a line of armed soldiers acts as the event which triggers the discharge of weapons.[29] In any other situation in which the imperative voice is used, the anticipated response is action rather than verbal expression.

Lotus's organization of command terms into the menu command hierarchy, may contain expression, but it also serves a utilitarian role, it commands the program to perform an operation.

---

[28] U.C.C. §3-110(1).

[29] Some words are so imbued with action that they are themselves treated as conduct. The classic case in First Amendment law is the treatment of "fighting words." These are "personally abusive epithets which, when addressed to the ordinary citizen, are, as a matter of common knowledge, inherently likely to provoke violent reaction." *Cohen v. California*, 403 U.S. 15, 20 (1971). See also *R.A.V. v. City of St. Paul*, 505 U.S. 377, 381 (1992).

III

Conclusion

Section 102(b) requires that a computer menu command hierarchy be denied copyright protection. Such a hierarchy as used operates as a *pure command function*. Only if the expressive aspect can be separated from the function of command can that separated aspect alone be protected by copyright. The menu hierarchy in this case should accordingly be denied copyright protection.

Respectfully submitted,

CAROL A. KUNZE
50 Palm Ave.
San Francisco, California
94118
(415) 386-1785

HOWARD C. ANAWALT
Counsel of Record
University of Santa Clara
Santa Clara, California
95053
(408) 554-4090

Counsel for Amicus Curiae

November 29, 1995

BEST AVAILABLE COPY

TABLE OF CONTENTS

Page

TABLE OF CONTENTS i

TABLE OF AUTHORITIES iii

INTEREST OF AMICI 1

SUMMARY OF ARGUMENT 3

ARGUMENT 4

I. THE FIRST CIRCUIT'S DECISION FALLS SQUARELY WITHIN THE MAINSTREAM OF RECENT SOFTWARE COPYRIGHT DECISIONS, AND THUS IS CONSISTENT WITH THIS COURT'S DECISION IN *FEIST* 7

II. LOTUS MISREADS THE FIRST CIRCUIT'S OPINION 12

III. THE FIRST CIRCUIT CORRECTLY CONCLUDED THAT COPYRIGHT PROTECTION DOES NOT EXTEND TO THE LOTUS 1-2-3 COMMAND STRUCTURE 15

A. Lotus Understates the Functionality of the 1-2-3 Command Structure 16

B. The First Circuit Correctly Recognized that Compatibility Considerations Effectively Limited the Alternatives Available to Borland 20

C. Judge Boudin Correctly Identified the Adverse Consequences of Extending Copyright Protection to the 1-2-3 Command Structure 23

Page

D. Lotus and Its Amici Grossly Exaggerate the Adverse Consequences, If Any, of the First Circuit's Decision 25

CONCLUSION 29

TABLE OF AUTHORITIES

Page

CASES

In re Alappat, 33 F.3d 1526 (Fed. Cir. 1994) 28

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995) 11

Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) 20

Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992) 11, 21

Baker v. Selden, 101 U.S. 99 (1879) 3, 11

Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989) 11

Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164 (1994) 25

Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2nd Cir. 1992) ..... *passim*

Diamond v. Diehr, 450 U.S. 175 (1981) 28

Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), *opinion supplemented on denial of reh'g en banc*, 46 F.3d 408 (5th Cir. 1995) 11

Engineering Dynamics, Inc. v. Structural Software, Inc., 46 F.3d 408 (5th Cir. 1995) 22

Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340 (1991) ..... *passim*

Gates Rubber Co. v. Bando American, Inc., 798 F. Supp. 1499 (D. Colo. 1992), *aff'd in part, vacated in part*, 9 F.3d 823 (10th Cir. 1993) 10

Gates Rubber Co. v. Bando Chem. Indus. Ltd., 9 F.3d 823 (10th Cir. 1993)..... 11

Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass 1992), *rev'd*, 49 F.3d 807 (1st Cir.), *and cert. granted*, 116 S. Ct. 39 (1995)..... 22

Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir.), *and cert. granted*, 116 S. Ct. 39 (1995)..... 18

Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir.), *cert. granted*, 116 S. Ct. 39 (1995)..... *passim*

Morrissey v. Procter & Gamble Co., 379 F.2d 675 (1st Cir. 1967) 13

Qualitex Co. v. Jacobson Prod. Co., 115 S. Ct. 1300 (1995)..... 21

Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987)..... 8

Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992)..... 11, 21

United States v. Microsoft Corp., 56 F.3d 1448 (D.C. Cir. 1995) 28

Whelan Associates, Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987)..... *passim*

STATUTES & REGULATIONS

17 U.S.C. § 101 5
17 U.S.C. § 102(a)..... 29
17 U.S.C. § 102(b) ..... *passim*
35 U.S.C. § 102 29
35 U.S.C. § 103 29
35 U.S.C. § 112 29
60 Fed. Reg. 28,778 (1995)..... 28

OTHER AUTHORITIES

J. Band and M. Katoh, *Interfaces on Trial* (1995).... 5, 21

R. Frank and P. Cook, *The Winner-Take-All Society* (1995) 28

M. Knell, *Microsoft Fights for Intuit*, The Boston Herald, Apr. 28, 1995..... 18

P. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045 (1989)..... 24

C. Morris and C. Ferguson, *How Architecture Wins Technology Wars*, Harv. Bus. Rev. (Mar./Apr. 1993) 18

M. Nimmer and D. Nimmer, *Nimmer On Copyright* (1995) 8

President's Information Infrastructure Task Force, *Global Information Infrastructure: Agenda for Cooperation* (U.S. Government Printing Office, Washington, D.C., Feb. 1995)..... 24

G. Saloner, *Economic Issues in Computer Interface Standardization*, 1 Econ. Innov. New. Tech. 135 (1990) 27

J. Sandberg and G. Hill, *Microsoft Probe Spurs Subpoenas Tied to Internet*, Wall St. J., Dec. 4, 1995 7

J. Sandberg, *Sun and Netscape Are Forming Alliance Against Microsoft on Internet Standard*, Wall St. J., Dec. 4, 1995 7

U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (U.S. Government Printing Office, Washington, D.C., May 1992) 6, 19

F. Warren-Boulton, K. Baseman & G. Woroch, *Copyright Protection of Software Can Make Economic Sense*, 12 Computer Law. 10 (Feb. 1995) 27

BRIEF AMICI CURIAE OF AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS AND COMPUTER & COMMUNICATIONS INDUSTRY ASSOCIATION IN SUPPORT OF RESPONDENT

The American Committee for Interoperable Systems and the Computer & Communications Industry Association submit this brief as *amici curiae* and respectfully request that the Court affirm the decision below. The letters of consent of Petitioner and Respondent are being filed with this brief.

INTEREST OF AMICI

The American Committee for Interoperable Systems ("ACIS") is an informal organization of companies that develop innovative software and hardware products that interoperate with computer systems developed by other companies.[1] Computer & Communications Industry Association ("CCIA") members participate in many sectors of the com-

---

[1] ACIS members include Accolade, Inc., Advanced Micro Devices, Amdahl Corporation, America Online, Inc., AT&T Global Information Solutions, Broderbund Software, Inc., Bull HN Information Systems, Inc., Chips and Technologies, Inc., Clearpoint Research Corporation, Color Dreams, Inc., Comdisco, Inc., Emulex Corporation, Forecross Corporation, The Fortel Group, Fujitsu Systems Business of America, Inc., Hitachi Data Systems, ICTV, Insignia Solutions, Integrated Documents Applications Corp., Johnson-Laird, Inc., Landmark Systems Corporation, LCS/Telegraphics, MidCore Software, Inc., New York Systems Exchange, Inc., Octel Communications Corporation, Phoenix Technologies, Ltd., Plimoth Research Inc., QAD Inc., Seagate Technology, Inc., Software Association of Oregon, Storage Technology Corporation, Sun Microsystems, Inc., Tandem Computers, Inc., 3Com Corporation, Trilium Consumer Electronics, Inc., Western Digital Corporation, and Zenith Data Systems Corporation. (The Software Association of Oregon consists of over 550 software development firms, firms in associated industries, and individuals professionally involved in software development.)

puter and telecommunications industry and range in size from small entrepreneurial firms to the largest in the industry.[2] Collectively, ACIS and CCIA members generate revenues in excess of $180 billion and employ over one million people.

ACIS and CCIA believe that computer programs deserve effective intellectual property protection to give developers sufficient incentive to create new programs. At the same time, ACIS and CCIA are concerned that the improper extension of copyright law will impede innovation and inhibit fair competition in the computer industry. ACIS and CCIA seek the application of legal standards that will effectuate copyright law's fundamental aims by ensuring authors "the right to their original expression," but also encouraging competitors "to build freely upon the ideas and information conveyed by a [copyrighted] work." *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349-50 (1991).

Neither ACIS, CCIA, nor their members have a direct financial interest in the outcome of this litigation. However, reversal of the First Circuit's decision would have serious anti-competitive consequences for ACIS and CCIA members and the computer industry as a whole. Specifically, extending copyright protection to elements necessary for interoperability would inhibit the ability of ACIS and CCIA companies to develop innovative, competitive products. ACIS and CCIA are filing this brief in order to address the

---

[2] CCIA members include Amdahl, AT&T, Bell Atlantic, Cambridge Technology Partners, Charles River Data Systems, Datum, Inc., Formation, Fujitsu Network Switching, Hitachi Data Systems, InterCAP Graphic Systems, Leasing Solutions, Northern Telecom, NYNEX, Okidata, Storage Technology Corporation, Summa Four, Inc., Sun Microsystems, Inc., Tandem Computers, Inc., TSI International Software Ltd., and ViON Corporation.

important intellectual property and competition law policy issues at stake in this case.

SUMMARY OF ARGUMENT

The decisive issue in this case is whether copyright law can protect the rules that enable two elements of a computer system to work together. To be sure, Lotus and its *amici* talk about other matters: the protectability of a computer program's user interface; the protectability of source code; and the claimed methodological flaws in the First Circuit decision. But, boiled down to its essence, the only question for decision by this Court is whether copyright can protect the rules that enable one computer program to connect with another.

This brief first addresses Lotus' mistaken claim that, at odds with other circuits, the First Circuit fashioned a special copyright rule for software. In fact, the First Circuit followed the rule applied in other circuits — a rule mandated by this Court for over a century, from *Baker v. Selden*, 101 U.S. 99 (1879) to *Feist* — that requires a thinner scope of copyright protection for utilitarian literary works than for novels and plays. The brief next rebuts Lotus' assertion that the First Circuit found separable expression in its command structure, but withheld protection from this expression because it was included in an unprotectable method of operation. Lotus misreads the First Circuit's decision. The First Circuit found no separable expression; rather, it held that the expression had *merged* with the method of operation.

The brief then turns to the decisive issue in this case: the protectability of interface specifications, and not the protectability of the great bulk of program elements that, under the First Circuit's decision and this Court's rulings, will continue to be protected by copyright. The 1-2-3 command structure is more than a user interface; it is the interface

between the Lotus program and programs — referred to as "macros" — that are written by users at their own considerable expense for execution in connection with the 1-2-3 program. Because the 1-2-3 command structure provides the template for the macros and because the macros are the key to compatibility, the First Circuit, consistent with holdings in other circuits, ruled that those elements necessary to macro compatibility are not protected by copyright. This ruling will not harm innovation; indeed, by promoting competition, in products that differ — transform — the original, it will lead to increased innovation.

ARGUMENT

Unlike traditional literary works such as novels and plays that stand alone and do not need to interact with any other work, computer programs never function alone; they function only by interacting with the computer environment in which their developers place them. This environment is absolutely unforgiving. Unless the computer program conforms to the precise rules for interacting with the other elements of the system, no interaction between the program and the system is possible. As a consequence, no matter how much better or cheaper the new program is, it will not enjoy a single sale if it cannot interoperate in its intended environment. If the developer of one part of the environment can use copyright law to prevent other developers from writing programs that conform to the system of rules governing interaction within the environment — interface specifications, in computer parlance — the first developer could gain a patent-like monopoly over the system without ever subject-

ing it to the rigorous scrutiny of a patent examination. Lotus seeks to use copyright in exactly this manner.[3]

Neither the Lotus Brief to this Court nor the briefs of its two amici[4] explain what is really at stake in this case, or why Lotus' position has so little support in the software industry. Indeed, the briefs of Lotus and its amici leave the misimpression that the few companies supporting the Petitioner do so out of concern that the First Circuit's decision will undermine their ability to keep others from copying their code, *i.e.*, the actual lines of "instructions" used "in a computer to bring about a certain result." 17 U.S.C. § 101 (definition of computer program); *see, e.g.*, AIPLA Brief at 13; Lotus Brief at 49.

ACIS and CCIA members rely on the copyright law to prevent unauthorized copying of the protectable expression contained in their code. Had the First Circuit's opinion called into question the protection of code, Lotus would have support from far more than the four companies that have filed a brief on its behalf. In truth, there is no expectation in the industry that the First Circuit's decision will endanger the scope of copyright protection for code in any respect.

---

[3] IBM (Lotus' parent) and *amicus* Intel both have also used copyright law to protect and extend their market power. *See* J. Band and M. Katoh, *Interfaces on Trial* (1995) at 26-28, 35-39.

[4] The American Intellectual Property Law Association ("AIPLA") Brief is hardly a brief "in support of neither party." AIPLA Brief at 1. It makes virtually the same arguments as the Lotus Brief and the joint Digital Equipment, Gates Rubber, Intel and Xerox brief ("DEC Brief"). This comes as no surprise, given that one of the law firms on the AIPLA brief represents a party that would benefit from a reversal of the First Circuit's decision in a case pending in the Eleventh Circuit, *Mitek Holdings, Inc. v. Arce Engineering Co.*, No. 94-5262 (11th Cir. filed Nov. 30, 1994) (appeal from 864 F. Supp. 1568 (S.D. Fla. 1994)).

Lotus and its few amici press this case not out of concern for code copying, but rather in an effort to extend the copyright law to prevent the emergence of compatible products that compete with their own. This case is about whether competitors can introduce compatible products that emulate, as they must, interface specifications, the rules that form the "external design" of a program.[5] This case is an attempt by a few companies who are the "first comers" to particular markets to use copyright law to preclude competitors from using the same *external* design, even though it is implemented in *wholly original* program code.

ACIS and CCIA believe such compatibility, or interoperability as it is sometimes called, is beneficial to both the industry and to the public. A program is compatible or interoperable with another program when it can interact with other programs in the same way as the first program. Borland's Quattro Pro program is compatible with Lotus 1-2-3 because it interacts with other programs in the same way — both programs can run "on top of" the same operating system (MS DOS) and both programs execute the same application macros written by users and other third-parties.

Compatibility is achieved by conforming to the rules that the developer establishes as a "socket" to enable another program to "plug into" it. Extending copyright to such rules has economic consequences far broader than what was contemplated by Congress in protecting the instructions used in a computer to bring about a certain result. In concrete terms, reversal of the First Circuit's decision would threaten a large and vibrant sector of the U.S. software industry that depends

---

[5] U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (U.S. Government Printing Office, Washington, D.C., May 1992) ("OTA Report") at 26.

on its ability to interoperate with computer systems developed by other firms. Copyright protection for interface specifications would lead to monopolies within each product market in the industry. Indeed, unprecedented economic concentration would result if one firm succeeds in asserting proprietary control over a critical interface specification in the information superhighway.[6]

I. THE FIRST CIRCUIT'S DECISION FALLS SQUARELY WITHIN THE MAINSTREAM OF RECENT SOFTWARE COPYRIGHT DECISIONS, AND THUS IS CONSISTENT WITH THIS COURT'S DECISION IN *FEIST*.

Lotus devotes much of its argument to proving the obvious: that computer programs are literary works under the Copyright Act and, as such, are governed by the same principles as other literary works. Lotus suggests that by considering the 1-2-3 command structure's utilitarian nature, the First Circuit somehow departed from congressional intent and from precedent in other circuits. Lotus Brief at 18. Lotus fails to mention, however, that although all literary works are governed by the same principle — specifically the idea/expression distinction embodied in Section 102(b) of the Copyright Act — this principle dictates that utilitarian works receive a thinner scope of protection than works such as novels or plays. Indeed, this Court explicitly recognized this point in *Feist*.

The appellate software copyright decisions since *Feist* all acknowledge that utilitarian works enjoy only a thin scope of protection. Yet, prior to *Feist*, the courts headed in the

---

[6] *See, e.g.*, J. Sandberg and G. Hill, *Microsoft Probe Spurs Subpoenas Tied to Internet*, Wall St. J., Dec. 4, 1995; J. Sandberg, *Sun and Netscape Are Forming Alliance Against Microsoft on Internet Standard*, Wall St. J., Dec. 4, 1995.

opposite direction. *Dicta* in one of the earlier decisions, *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987), signaled extremely broad protection. The *Whelan* court defined the unprotectable idea in the program at issue to be "to run a dental laboratory in an efficient way," 797 F.2d at 1238 n.34, while the protectable expression was "the manner in which the program operates." *Id.* at 1239. Subsequent courts and commentators interpreted the *Whelan dicta* as drawing the line between idea and expression at a very high level of abstraction, suggesting the availability of copyright protection for virtually everything in a computer program except the most general statement of its function.[7] Indeed, some courts and commentators refer to a "*Whelan* rule" that a computer program contains but one idea.[8] *Id.*

Courts in other circuits quickly rejected the approach taken in *Whelan's dicta*. The Court of Appeals for the Fifth Circuit recognized that although computer programs are literary works, they are utilitarian and thus are strictly constrained by factors external to the program, such as industry standards. *Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc.*, 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987). Accordingly, the *Plains Cotton* court explicitly "decline[d] to embrace *Whelan*." *Id.*

The rejection of *Whelan* accelerated after the *Feist* decision. Thus, the Court of Appeals for the Second Circuit in

---

[7] *See, e.g.*, 3 M. Nimmer & D. Nimmer, *Nimmer On Copyright*, § 13.03[F] (1995); *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2nd Cir. 1992).

[8] The *Whelan* court also held that copyright protects the expression in the non-literal elements of a computer program. This holding, a simple restatement of the fundamental copyright principle that copyright protects original expression in works of authorship, is non-controversial and has been readily accepted by other courts.

Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 705 (2nd Cir. 1992) observed that "*Whelan's* general formulation that a program's overall purpose equates with the program's idea is descriptively inadequate."

The Second Circuit also challenged the two justifications the *Whelan* opinion had given for drawing the idea/expression line at a high level of abstraction. First, the *Whelan* court had observed that computer programs are "literary works," and that copyright in other types of literary works is infringed if competitors use the works' structure and organization. The *Altai* court acknowledged that computer programs were "literary works," but at the same time recognized the "essentially utilitarian nature of a computer program," and emphasized that, "compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in Section 102(b)." *Id.* at 704.

Second, the *Whelan* court had noted that "among the more significant costs in computer programming are those attributable to developing the structure and logic of the program" and that protecting the structure and logic "would provide the proper incentive for programmers by protecting their most valuable efforts ...." *Whelan*, 797 F.2d at 1237. The *Altai* court rejected *Whelan's* incentive-based justification for broad copyright protection as having "a corrosive effect on certain fundamental tenets of copyright doctrine." *Altai*, 982 F.2d at 712. The Second Circuit explained that

> [t]he interest of the copyright law is not in simply conferring a monopoly on industrious persons, but in advancing the public welfare through rewarding artistic creativity, in a manner that permits the free use and development of non-protectable ideas and processes.

Id. at 711.

Significantly, the two foregoing points flowed directly from this Court's decision in *Feist.* With respect to the argument that a computer program is a literary work, the Court made clear that merely calling a work a "literary work" does not determine the scope of its copyright protection and that not all literary works are entitled to the same scope of protection: "the copyright in a factual compilation is thin"; "[t]his Court has long recognized that the fact/expression dichotomy limits severely the scope of protection in fact-based works." *Feist*, 499 U.S. at 349, 350. Similarly, with respect to the incentive-based justification, "*Feist* teaches that substantial effort alone cannot confer copyright status on an otherwise uncopyrightable work .... Thus, *Feist* implicitly undercuts the *Whelan* [incentive based] rationale ...." *Altai*, 982 F.2d at 711.

Whelan's dicta had led some lower courts to conclude wrongly that "the idea-expression distinction is somewhat diluted in analysis applied in infringement cases concerning computer programs." *Gates Rubber Co. v. Bando American, Inc.*, 798 F. Supp. 1499, 1510 (D. Colo. 1992), *aff'd in part, vacated in part*, 9 F.3d 823 (10th Cir. 1993). In contrast, the *Altai* court correctly recognized that because "[t]he essentially utilitarian nature of a computer program ... complicates the task of distilling its idea from its expression," *Altai*, 982 F.2d at 704, the idea/expression dichotomy must be applied in software copyright cases with great rigor and extreme care.

Every appellate software copyright decision since *Altai* agrees with its fundamental teachings: although computer programs are literary works protected by copyright, and often require great expense and effort to develop, as utilitarian works they are entitled to "thinner" protection than novels or plays; and to ensure that they do not receive too much

protection, the courts must closely scrutinize the works at issue and apply the idea/expression dichotomy meticulously. *See, e.g., Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *Apple Computer, Inc. v. Microsoft Corp.*, 25 F.3d 1435, 1439 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995); *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1347 n.12 (5th Cir. 1994), *opinion supplemented on denial of reh'g en banc*, 46 F.3d 408 (5th Cir. 1995); *Gates Rubber Co. v. Bando Chemical Indus. Ltd.*, 9 F.3d 823, 836 (10th Cir. 1993); *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992).

Underlying these appellate decisions is not only this Court's decision in *Feist*, but a long line of Supreme Court decisions cautioning against the improper extension of the patent laws via copyright law or state law. As the Court stated in *Baker v. Selden*, 101 U.S. 99, 102 (1879), "[t]o give the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public." More recently, this Court in *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 146 (1989), condemned state law protection for inventions left by the Patent Act in the public domain: "the Federal patent laws have embodied a careful balance between the need to promote innovation and the recognition that imitation and refinement through imitation are both necessary to invention itself and the very lifeblood of a competitive society." The software copyright appellate decisions, by refusing to treat software functionality as protected expression, avoid upsetting this careful balance. *See, e.g., Atari*, 975 F.2d at 839.

The First Circuit's decision falls squarely within this body of case law. Judge Boudin's concurring opinion clearly recognized that software, although a literary work, receives a different scope of protection from novels, plays, and films:

"Utility did not bar copyright (dictionaries may be copyrighted), but it alters the calculus." *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 820 (1st Cir.), *cert. granted*, 116 S. Ct. 39 (1995). Further, Judge Boudin does not "erect special judicial barriers to protection" of software. DEC Motion for Leave to File Amicus Curiae Brief at 3. Rather, like the other circuit court decisions cited above, he believed that Congress intended courts to apply Section 102(b) of the Copyright Act in a case-by-case manner, and the thinner protection for software, like the thinner protection for other utilitarian literary works such as dictionaries, flows directly from Section 102(b). He also recognized that failure to apply Section 102(b) strictly in software copyright cases would cause copyright to infringe the jurisdiction of the patent laws.

Further, both Judge Stahl's and Judge Boudin's opinions reflect a thorough consideration of the program element at issue and its relationship to its environment, in an effort to understand on which side of the idea/expression divide it should fall. This willingness to grapple with complex technology, rather than succumb to the temptation of superficial understanding, makes the First Circuit's decision a model other courts should follow.

II. LOTUS MISREADS THE FIRST CIRCUIT'S OPINION.

In addition to faulting the First Circuit for finding that copyright accords software thinner protection than novels and plays, Lotus attacks the First Circuit's application of Section 102(b) to the facts of this case. Lotus contends that the First Circuit agreed with the district court below that the Lotus 1-2-3 command structure contained expression separable from its ideas and functionality, but that the First Circuit nonetheless refused to extend copyright protection to this expression because it was part of the program's method of operation. The First Circuit committed this error, Lotus

claims, by relying on a flawed definition of "method of operation," which led it to take a "dangerous and improper shortcut in applying § 102(b)." Lotus Brief at 18; AIPLA Brief at 7.

The linchpin of this argument is Lotus' view that the First Circuit conceded that the command structure contained *separable* expression. In fact, the First Circuit made no such concession. To be sure, it stated that "the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms ...." 49 F.3d at 816. But the First Circuit made it quite clear that these expressive choices had *merged* with the command structure. Thus, two sentences after its reference to Lotus' expressive choices, the First Circuit stated that "[i]f *specific* words are *essential* to operating something, then they are part of a 'method of operation' and, as such, are unprotectable." *Id.* (emphasis supplied).

Similarly, in its discussion of the VCR analogy, the First Circuit stated that "the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves." *Id.* at 817. Pursuing this thought, the court stated that "[u]nlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are *essential* to operating Lotus 1-2-3." *Id.* (emphasis supplied). The First Circuit underscored its conclusion that the developers' expressive choices had merged with the command structure, the method of operating the program, by citing the First Circuit's venerable decision in *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967), for the proposition that "[w]hen there are a limited number of ways to express an idea ... the expression '*merges*' with the idea and becomes uncopyrightable." 49 F.3d at 818 n.13 (emphasis supplied).

Because it concluded that the expressive terms had merged with the command structure, the question before the First Circuit — and the real question before this Court — was whether the command structure *itself* was protected expression or unprotected idea.[9] Lotus and its *amici* suggest that by not applying the *Altai* abstraction-filtration-comparison test, the First Circuit failed to assess properly whether the command structure should receive copyright protection. *See* Lotus Brief at 40; AIPLA Brief at 7. This argument, too, is predicated on a misreading of the First Circuit's opinion.

The First Circuit reasoned that the *Altai* test might be "misleading" if applied here because it could cause the identification of expression at a low level of abstraction while obscuring the fact that the expression may be an essential part of the method of operation at a higher level of abstraction. The First Circuit's reasoning is sound where, as here, there is verbatim copying of a specific element; further abstraction would simply reveal similarities at progressively more detailed levels, while deconstructing the overall system into which the sub-elements had merged. In this case, one would end up with hundreds of identical terms, without any sense that those terms were an essential part of a larger whole.

Nonetheless, it is important to note that the First Circuit did not dispense with all three steps of the *Altai* test, just the first one — abstraction. It still performed the critical second step — filtration — on the command structure. As the next

---

[9] Prior to the Argument section of its Brief, Lotus appears to agree with this formulation of the issue: "It is the copyrightability of the overall combination of words and menus in the 1-2-3 menu command hierarchy, viewed as a whole, and *not* any individual menu command ... viewed in isolation, that is at issue in this case." Lotus Brief at 6 (footnote omitted).

section of the brief explains, the First Circuit performed the filtration step properly.

III. THE FIRST CIRCUIT CORRECTLY CONCLUDED THAT COPYRIGHT PROTECTION DOES NOT EXTEND TO THE LOTUS 1-2-3 COMMAND STRUCTURE.

Lotus properly states that "Section 102(b) requires courts to attempt to separate the purely functional attributes of computer programs from the particular expression chosen by the programmer to accomplish or to provide that functionality." Lotus Brief at 38. A few sentences later, Lotus restates this proposition as follows: "the expression is protected by copyright if it is only one of numerous possibilities for providing the same functionality." *Id.* at 39.

These two statements get right to the heart of the matter: Lotus and the First Circuit have different meanings for the phrase "same functionality." For Lotus, a command structure with the "same functionality" as the 1-2-3 command structure is one that offers "exactly the same selection of functions as does 1-2-3" but with different terms in a different sequence. *Id.* at 41. For the First Circuit, a command structure with the "same functionality" as the 1-2-3 command structure is one that is completely 1-2-3 compatible. If Lotus' definition of "same functionality" is correct, then the 1-2-3 command structure is but one of many ways of expressing the same functionality. Conversely, if the First Circuit's definition of "same functionality" is correct, then the 1-2-3 command structure is essential to achieving the same functionality. Thus, the core inquiry before this Court, and the issue on which this case turns, is whether compatibility is a functional characteristic. Below, we will explain why compatibility unquestionably is a functional characteristic unprotected by copyright. But before we do so, we must explain the nature of the product at issue.

A. Lotus Understates the Functionality of the 1-2-3 Command Structure.

Most plaintiffs in copyright infringement cases tend to exaggerate the significance of their work in an effort to magnify the work's expressive quality. Here, by contrast, Lotus down plays the nature of its command structure in an effort to minimize the structure's inherent functionality. Throughout their briefs, Lotus and its amici repeatedly refer to the 1-2-3 command structure as a "user interface" which "communicate[s] the product's underlying functionality to users in a clearly organized presentation." Lotus Brief at 9 (citations omitted).

The command structure, however, has a second function which Lotus relegates to footnotes but which in fact is critical to both the Lotus product and this case: it functions as the interface between the spreadsheet program and "macros," which are programs written by users with the 1-2-3 commands. It is this program-to-program interface function that is of primary concern to ACIS and CCIA.

A consideration of the various kinds of programs that run on modern computers will clarify this function of the command structure. All computers must run an "operating system" program, which is the fundamental program that instructs the computer how to operate, how to manage its resources, and how to run other programs. *Altai*, 982 F.2d at 698. Common operating systems include Windows 95 for IBM-compatible personal computers and UNIX for workstations. The operating system serves as a "platform" for "application programs" such as word processing programs, database programs, multimedia games, or, as in this case, spreadsheet programs. Application programs must conform precisely to the functional requirements — the interface specification — of the operating system, or else the applications will not work properly.

Although Lotus 1-2-3, taken as a whole, is an application program, it assumes some of the characteristics of an operating system with respect to the user written application programs — the macros — that attach to it. In other words, Lotus 1-2-3 serves as a platform upon which the macros run. The command structure defines the rules for interaction between the 1-2-3 macros and the Lotus platform, and thus acts as an interface specification. Put differently, the command structure provides the syntax and semantics of the communication between a 1-2-3 macro and the Lotus platform.

Many Lotus users have invested substantial time and resources developing libraries of highly complex customized macros appropriate to their business needs. Indeed, the Lotus users collectively have invested far more resources in the development of their macros than Lotus ever invested in the development of Lotus 1-2-3. Lotus claims that its programmers "spent hundreds of hours over a period of many months" developing the 1-2-3 command structure. Lotus Brief at 9. Lotus users, in contrast, collectively have spent thousands, if not millions, of hours over many years writing their macros. Because of their investment in the macros, the Lotus users are "locked-in" to the Lotus environment: as they expand their operations, they simply will not purchase spreadsheet programs developed by a Lotus competitor such as Borland unless the Borland spreadsheet can also execute their macros.

The most basic form of macro compatibility requires the Borland platform to have the ability to translate the macro's instructions into instructions intelligible to the Borland platform, and vice versa. Because the set of instructions used by the macro is a subset of Lotus' commands, the Borland platform had to translate those instructions from the macros by means of a file that exactly replicated the Lotus 1-2-3 command structure's syntax and semantics. Borland calls

this file the "Key Reader."[10] *Lotus v. Borland*, 49 F.3d at 811-12.

Complete macro compatibility, however, requires more than just the ability to run existing macros. Users of the Borland platform must be able to correct errors in the macros, and modify the macros to address new problems. Because the macros are constructed out of Lotus commands, the user of the Borland platform can debug or modify the macros only if the platform can display the commands on the terminal and only if the platform can understand the Lotus commands keyed in by the user. Thus, to perform the debugging and modification functions essential for complete macro compatibility, the Borland platform must reproduce the Lotus command structure. *Id.* at 811 n.3.

---

[10] In this context, it is worth noting that Microsoft's spreadsheet product Excel at one time also had a file like Borland's Key Reader which permitted 1-2-3 macro compatibility. Lotus did not challenge Microsoft in court, but instead permitted Microsoft's product to remain compatible with 1-2-3 while Borland was precluded by court injunction from offering a compatible product. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223, 229-30, 234 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir.), *and cert. granted*, 116 S. Ct. 39 (1995).

The reason Excel ultimately eclipsed Lotus 1-2-3 is a further demonstration of the significance of interface specifications and compatibility: Microsoft introduced a new operating system (Windows), migrated the installed base of computer users from the old operating system (DOS) to the new one (Windows), and released a Windows-compatible version of Excel before providing Lotus with sufficient interface information to permit the release of 1-2-3 for Windows. *See* C. Morris and C. Ferguson, *How Architecture Wins Technology Wars*, Harv. Bus. Rev. (Mar./Apr. 1993). Indeed, Lotus vociferously complained to the Antitrust Division of the Department of Justice about Microsoft's conduct in manipulating the forces of the free market to achieve a monopoly. *See, e.g.*, M. Knell, *Microsoft Fights for Intuit*, The Boston Herald, Apr. 28, 1995, at 23.

Programmers view an interface specification such as the 1-2-3 command structure as part of the external design of a program. As the Office of Technology Assessment noted,

> [p]rograms have an external design or interface — the conventions for communication between the program and the user or other programs. The external design is conceptually separate from the program code that implements the interface (the internal design). It specifies the interactions between the program and the user or other programs, but not how the program does the required computations. There are typically many different ways of writing a program to implement the same interface.

OTA Report at 26. Using copyright jargon, it can accurately be said that the interface specification is the "system" or "method of operation" that is "expressed" by the program code.

In this case, the 1-2-3 command structure was the interface specification for the Lotus spreadsheet. Lotus designed its program so that macros had to be written with the terms and sequence of the command structure in order to operate with the 1-2-3 program. Borland subsequently entered the market seeking to compete against Lotus for the business of users who had already written macros which used the command structure as an *interface specification*. In order to run these third party macros, Borland adopted the same interface specification (the command structure) but implemented that specification in its own wholly original code.

B. The First Circuit Correctly Recognized that Compatibility Considerations Effectively Limited the Alternatives Available to Borland.

Unlike the district court below, the First Circuit appreciated the importance of macro compatibility, and this appreciation strongly influenced the First Circuit's perception of the command structure: "That the Lotus menu command hierarchy is a 'method of operation' becomes clearer when one considers program compatibility." 49 F.3d at 817. After observing that a user-written 1-2-3 macro could not operate with an incompatible spreadsheet program, the First Circuit concluded: "As the Lotus menu command hierarchy serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a 'method of operation.'" *Id.* at 818.

Lotus nonetheless contends that Borland sought to be compatible with the 1-2-3 macros "for commercial reasons, not as a technical necessity." Lotus Brief at 12. This argument echoes the suggestion of the Court of Appeals for the Third Circuit over a decade ago that compatibility was a "commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged." *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

More recently, however, other circuits have replaced "metaphysical distinctions" with "practical considerations." *Altai*, 982 F.2d at 706. They understand that software copyright issues do not arise in a vacuum, but in a particular commercial context that as a *practical* matter restricts a programmer's range of possible expression. In other words, these courts have acknowledged that a work's commercial

context helps define the boundary between idea and expression.[11] Thus, the Second Circuit in *Altai* recognized that

> a programmer's freedom of design choice is often circumscribed by extrinsic considerations such as (1) the mechanical specifications of the computer on which a particular program is intended to run; (2) *compatibility requirements of other programs with which a program is designed to operate in conjunction*; (3) computer manufacturers' design standards; (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry.

982 F.2d at 709-10 (emphasis supplied, citations omitted). The Second Circuit ruled that copyright does not protect program elements dictated by such external factors, including compatibility. *Id.*

The Federal Circuit in *Atari*, citing *Altai*, stated that "[t]he court must filter out as unprotectable ... expression dictated by external factors (like the computer's mechanical specifications, compatibility with other programs, and demands of the industry served by the program) ...." 975 F.2d at 839. Likewise, the Ninth Circuit in *Sega* expressly recognized that computer programs "contain many logical, structural, and visual display elements that are dictated by ... external factors such as compatibility requirements and

---

[11] *See Interfaces on Trial* at 87-90. In the trademark context, this Court has explained that "'a product feature is functional,' and cannot serve as a trademark, 'if it is essential to the use or purpose of the article or if it affects the cost or quality of the article,' that is, if exclusive use of the feature would put competitors at a significant non-reputation related disadvantage." *Qualitex Co. v. Jacobson Products Co.*, 115 S. Ct. 1300, 1304 (1995) (quoting *Inwood Laboratories, Inc. v. Ives Laboratories, Inc.*, 456 U.S. 844, 850 n.10 (1982)).

industry demands. In some circumstances, even the exact set of commands used by the programmer is deemed functional rather than creative for purposes of copyright." 977 F.2d at 1524 (citations omitted).[12]

Lotus v. Borland presents one of those circumstances in which "the exact set of commands used by the programmer is deemed functional" because it is essential for compatibility between the macro and the platform. The fact that Borland could theoretically have created a spreadsheet program incompatible with the 1-2-3 macros (and ultimately did so in 1992) is irrevelant. The critical point is that *within* the 1-2-3 environment, users and would-be competitors have no alternative to the 1-2-3 command structure.

---

[12] *See also Engineering Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408, 410 (5th Cir. 1995), *supplementing opinion*, 26 F.3d 1335 (5th Cir. 1994) (copyright protection should not "extend to the manufacturing of computer hardware so as to deter achieving compatibility with other models"). The district court below stated that these appellate decisions simply stood for the proposition that copyright allows a new comer software developer to replicate the design of the first comer's "plug" in order to plug his software into the first comer's "socket," but that copyright nonetheless prohibits the new comer from replicating the design of the "socket" in order to satisfy the demand for additional outlets. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 212-13 (D. Mass. 1992), *rev'd*, 49 F.3d 807 (1st Cir.), *and cert. granted*, 116 S. Ct. 39 (1995). The district court reached this metaphysical conclusion by reasoning that the socket was created before the programs with the compatible plugs. The precise order of creation, however, should make little difference; if copyright does not protect the design of the plug, it surely does not protect the design of the socket. Moreover, even if the socket was actually constructed before the plugs, the socket and the plug obviously were designed at the same time so as to fit into one another.

C. Judge Boudin Correctly Identified the Adverse Consequences of Extending Copyright Protection to the 1-2-3 Command Structure.

In his concurring opinion, Judge Boudin observed that "[i]f Lotus is granted a monopoly on this pattern, users who have ... devised their own macros are locked into Lotus ...." 49 F.3d at 821. Judge Boudin further observed that granting a monopoly over a standard would have the additional undesirable effect of blocking development of a later, superior product:

> Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the *de facto* standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet — either in quality or in price — there may be nothing wrong with this advantage.
>
> But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance ....

Id. at 821.

ACIS and CCIA believe that Judge Boudin's reasoning applies to all interface specifications. The extension of copyright protection to such interface specifications would be wrong as a matter of copyright law, and would effectively

eliminate competition in operating systems, or any software product that functions as a platform for other software products.

Such a broad monopoly would have serious implications for consumer welfare.[13] In the absence of competition during the effective lifespan of the product, the first developer would have little incentive to develop more innovative and less costly products. These negative consequences would be compounded by the fact that the personal computer revolution, and recent major improvements in communications technology, have produced an overwhelming need for interconnection between different elements of computer systems. Within a given large corporation, literally thousands of personal computers and workstations scattered across the globe need to interact with each other and with the company's mainframes. Moreover, with the advent of the Global Information Infrastructure, different firms will need to exchange vast quantities of data through their computers.[14] Copyright protection for interface specifications would lock users into a particular operating system or network software environment, and would inhibit the transfer of data between users with different computing environments.

It should be stressed that compatible products are *not* mere "clones" that offer only the same functionality as the products of the first comer, but at a lower price. While compatible products must offer at least the same functionality, they typically offer additional functionalities not found in the first comer's products. Borland's Quattro Pro is an

---

[13] *See, e.g.*, P. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045, 1082, 1097 n.281 (1989).

[14] *See* President's Information Infrastructure Task Force, *Global Information Infrastructure: Agenda for Cooperation* (U.S. Government Printing Office, Washington, D.C., Feb. 1995) at 14-16.

excellent example of this transformation. While Quattro Pro included a 1-2-3 mode to enable compatibility with 1-2-3 macros, it also offered users a native Borland mode completely different from 1-2-3. Moreover, even in the 1-2-3 mode, Quattro Pro inserted additional commands not present in the 1-2-3 command structure.

The compatible products developed by ACIS and CCIA members all add value to consumers in a similar manner. They compete with the first comers' products not only in terms of price (indeed, sometimes the compatible products may be more expensive), but also in terms of innovation. In this respect, compatible developers' use of preexisting interface specifications is a transformative use of the sort accredited by this Court in *Campbell v. Acuff-Rose Music, Inc.*, 114 S. Ct. 1164 (1994).

Some ACIS and CCIA companies are small start-ups with a handful of employees. Others are Fortune 500 companies. All need to interoperate with the computer systems developed by other firms in order to reach the market for their new products, which push the technological envelope. Reversal of the First Circuit's decision threatens to wipe out this vital sector of the U.S. software industry that has helped maintain our leadership position in global markets.

D. Lotus and Its Amici Grossly Exaggerate the Adverse Consequences, If Any, of the First Circuit's Decision.

Lotus states that "[t]he First Circuit's decision effectively nullifies the copyright protection Congress enacted for the expression in computer programs." Lotus Brief at 42. Lotus reaches this breathtaking conclusion by pursuing the following reasoning: "If, as the First Circuit held, anything that can be defined as part of a method of operation is ineligible for copyright protection, then by the same logic virtually every-

thing in a computer program — source code no less than textual menus on the screen — is unprotected." *Id.* at 18. As discussed above, the critical flaw in Lotus' reasoning is that the First Circuit did *not* hold that anything that can be defined as part of a method of operation is ineligible for copyright protection; rather, it held that anything that is an *essential* part of a method of operation is ineligible for copyright protection. The word "essential" makes all the difference in the world; it signals merger of idea and expression.

The First Circuit found that the selection and arrangement of the individual terms and menus had merged with the 1-2-3 command structure. The analysis then shifted to whether the command structure, taken as a whole, received copyright protection. The court concluded that it did not, in large measure because of its macro compatibility functionality.

Contrary to Lotus' assertion, the First Circuit's decision leaves many original program elements protected. A program's actual lines of code, and the structure of the code, remain protected to the extent they are not essential to achieving specific functionality. Here, Borland could achieve 1-2-3 compatibility using completely different code. Similarly, many features of a program's screen displays remain protected. Indeed, one can even imagine menu commands that might receive protection. If a menu had fanciful icons instead of words, for example, the individual icons might receive protection.

The Information Technology Industry Council, in its brief *amicus curiae* in support of Lotus' petition for a writ of *certiorari*, argued that the First Circuit's ruling "contains no limiting principle by which to differentiate the aspects of a program that are copyrightable from those that are not protectable." ITIC Brief at 11. In fact, the First Circuit's decision does contain such a limiting principle: those elements essential for programs to achieve the same functionality do not receive copyright protection. The First Circuit further holds that because compatibility is a functional characteristic, those elements necessary to achieve compatibility do not receive copyright protection. If these limiting principles are not clear enough from the face of the First Circuit's decision, then this Court should announce them unambiguously.

Lotus doubtlessly will complain that even a ruling restricted to compatibility elements will diminish software firms' incentive to innovate. To be sure, this ruling will increase the competition faced by first comers. But this competition will merely prevent first comers from appropriating the returns on the *users'* substantial investment in developing their own programs and files based on the first comers' products.[15]

Further, economists have demonstrated that in markets with strong network externalities, such as the market for software, the first comer reaps enormous competitive advantages from the establishment of a *de facto* standard interface specification.[16] Copyright should not compound this

---

[15] *See* F. Warren-Boulton, K. Baseman & G. Woroch, *Copyright Protection of Software Can Make Economic Sense*, 12 Computer Law. 10, 23 (Feb. 1995)

[16] *See* G. Saloner, *Economic Issues in Computer Interface Standardization*, 1 Econ. Innov. New. Tech. 135, 140 (1990):

> Because of the compatibility and network benefits, all else equal, a new user prefers a vendor with a larger total installed base of users. Thus installed bases have a tendency to be self-perpetuating: they provide the incentive for the provision of products (software and hardware) that is compatible with that in the installed based which in turn attracts new users to the installed base further swelling its ranks and increasing the incentive for the provision of even more complementary products.

"winner takes all" result by locking the gateway to competition.

Lotus argues that antitrust law, rather than copyright law, should address the first comer's anticompetitive behavior. Lotus Brief at 47. A copyright, however, is a legal monopoly created by the Copyright Act, and the appropriate scope of that monopoly should be determined in the first instance by reference to the checks and balances contained in the Copyright Act itself, rather than the Sherman Act.

In any event, Lotus will still have ample incentive to develop improved versions of 1-2-3 as well as new products. As noted above, copyright will protect its implementation of programs that conform to the 1-2-3 command structure or other interface specifications that it develops in the future.

Moreover, Lotus can seek patent protection for its command structures. In *Diamond v. Diehr*, 450 U.S. 175 (1981), this Court opened the gates to software patents. Recent Federal Circuit decisions have pushed the gates open even further. *E.g., In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).[17] Under the evolving case law, many aspects of computer interfaces, including many command structures such as the one at issue here, may be patentable subject matter. Indeed, even before the above legal developments, companies were able to obtain utility patents on menu command hierarchies. J.A. 842-46, 856-63 (portions of U.S. Patent Nos. 4,989,141 and 4,611,306).

See also United States v. Microsoft Corp., 56 F.3d 1448, 1452 (D.C. Cir. 1995); R. Frank and P. Cook, *The Winner-Take-All Society* (1995).

[17] *See also* U.S. Patent and Trademark Office, Proposed Examination Guidelines for Computer-Implemented Inventions, 60 Fed. Reg. 28,778 (1995).

If interface specifications were protected under the copyright laws, the public would not have the same safeguards against overbroad monopolies that exist under the Patent Act. The sole requirement for a copyright is that it be an original, expressive work of authorship. 17 U.S.C. § 102(a). By contrast, an invention must satisfy the novelty requirements of 35 U.S.C. § 102 and the non-obviousness requirements set forth in 35 U.S.C. § 103. Furthermore, the specification must contain a sufficiently detailed description of the invention to enable one skilled in the relevant art to practice it, and must disclose the best mode contemplated by the inventor for carrying out the invention. 35 U.S.C. § 112. If a patent applicant must satisfy these standards to obtain a 20-year monopoly on a method of operating a computer, it is difficult to understand why that same inventor should obtain copyright protection in the same subject matter for 75 years or longer, without even satisfying those standards.

ACIS and CCIA members rely on copyright protection for many aspects of computer programs, and consider it sufficient for its intended purposes. ACIS and CCIA members also frequently obtain patents when warranted. This includes hardware and software processes such as the "selection and arrangement of executable operations." ACIS and CCIA are deeply concerned, however, that a software company might obtain patent-like protection for functional characteristics of software that do not meet the rigorous standards of patentability, merely by claiming a copyright. The First Circuit's refusal to allow Lotus to employ copyright law to obtain a *de facto* patent is consistent with this Court's refusal to allow Bonito Boats to employ Florida state law to obtain a *de facto* patent.

CONCLUSION

In order to achieve compatibility with macros written by the third-party users, it was necessary for Borland to use the

words of the 1-2-3 command structure, and to display those words on the computer screen. In using these words, Borland reproduced only the minimum portions of Lotus 1-2-3 necessary to achieve macro-compatibility, and nothing more. Creators of computer programs such as Borland should not be prevented by copyright law from using as much of another program's utilitarian methods of operation as is necessary to enable their programs to work with other programs. This is needed to ensure interoperability across computer systems and networks, and interoperability is needed to promote consumer welfare on the terms contemplated by the Copyright Act. The First Circuit correctly recognized the force of these principles in the applicable law. Under the First Circuit's holding, copyright will provide a secure basis for continued innovation and competition in the software industry.

For the above reasons, the decision of the First Circuit should be affirmed. This Court should hold that because compatibility is a functional characteristic, those program elements necessary to achieve compatibility do not receive copyright protection.

Respectfully submitted,

EDWARD J. BLACK
COMPUTER & COMMUNICATIONS INDUSTRY ASSOC.
666 Eleventh Street, N.W.
Washington, DC 20001
(202) 783-0070

PETER M.C. CHOY
Counsel of Record
AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS
2550 Garcia Avenue
Mountain View, CA 94043
(415) 336-2482

PAUL GOLDSTEIN
STANFORD LAW SCHOOL
Stanford, CA 94305
(415) 723-0313

December 8, 1995

Attorneys for Amici

No. 94-2003

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL, INC.,

Respondent.

On Writ of Certiorari to the
United States Court of Appeals
for the First Circuit

BRIEF OF AMICUS CURIAE
LEAGUE FOR PROGRAMMING FREEDOM
IN SUPPORT OF RESPONDENT

Brian McGrath
Counsel of Record
Columbia Law School
435 West 116th Street
New York, NY 10027
212-854-8538/7946 (Fax)

Pamela S. Karlan
781 Massie Road
Charlottesville, VA 22903
804-924-7810/7536 (Fax)

Counsel for Amicus Curiae

BEST AVAILABLE COPY

TABLE OF CONTENTS

Table of Contents i

Table of Authorities ii

Interest of Amicus Curiae 1

Summary of Argument 2

Argument

I. The Court of Appeals Correctly Applied §102(b) of the Copyright Act 6

A. Critical Technical Distinctions Exist Between Computer Software and Other Literary Works 7

B. User Interfaces are Language Systems, or Methods of Operation, Excluded from Copyright by §102(b) 12

II. Petitioner's Interpretation Leads to Absurd and Undesirable Results 16

III. The Court of Appeals Properly Refused to Expand the Statutory Monopoly Without a Clear Statement of Congressional Intent to Alter the "Fundamental Axiom" of Copyright 19

Conclusion 21

TABLE OF AUTHORITIES

Pages

Cases

Baker v. Selden, 101 U.S. 99 (1879) 3, 15

Feist Publications, Inc. v. Rural Tel. Service Co., 499 U.S. 340 (1991) 6, 16, 18

Harper & Row, Publishers, Inc. v. Nation Enterprises, 471 U.S. 539 (1985) 6

Lotus Development Corp. v. Borland International, Inc., 831 F. Supp. 223 (D. Mass. 1993) 11

Sony Corp. v. Universal City Studios, 464 U.S. 417 (1984) 2, 19, 20

Constitutional and Statutory Provisions

Copyright Act of 1976, 17 U.S.C. §102(b) 2,passim

Pub. L. No. 96-517, 94 Stat. 3015 (1980) 8

Other Materials

James Joyce, *Finnegans Wake* (1939) 12, 13, 14

H.R. Rep. No. 94-1476 (1976) 6, 9

S. Rep. No. 94-473 (1975) 6

No. 94-2003

IN THE

Supreme Court of the United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

BRIEF OF *AMICUS CURIAE* LEAGUE FOR PROGRAMMING FREEDOM IN SUPPORT OF RESPONDENT

Interest of *Amicus Curiae*

The League for Programming Freedom is a private nonprofit membership organization whose members are computer scientists, lawyers, and others committed to encouraging the diffusion of knowledge and useful arts

by opposing unwarranted extensions of intellectual property principles in the domain of computer technology. The League has been particularly concerned since its formation in 1988 with issues of software patents and computer interface copyrights. The present case presents issues of central concern to the League and its members. All parties have consented to the filing of this brief. Letters to that effect have been lodged with the Court.

Summary of Argument

This case, like *Sony Corp.* v. *Universal City Studios, Inc.*, 464 U.S. 417 (1984), concerns an attempt to expand the reach of the statutory monopoly conveyed through copyright beyond the limits set by Congressional intent. Here, as in *Sony*, profound social consequences for millions of individual Americans are at stake. Petitioners ask this Court to accept a theory of copyright, said to be implicit in a statute written before the problems here at issue had been recognized, that would alter, fundamentally and to the public detriment, the development of the century's most significant technology. The Court of Appeals correctly concluded that, far from implicitly accepting petitioner's view of the scope of copyright, Congress had in §102(b) of the 1976 Copyright Act, 17 U.S.C. §102(b), excluded program-control and user interfaces to computer programs from copyright coverage as a species of "methods of operation."

Amicus urges this Court to affirm the decision of the Court of Appeals. The court below correctly synthesized the words of §102(b), the distinction between idea and

expression in the theory of copyright, and the new technology involved. The interpretation of the 1976 Act given by the Court of Appeals harmonizes existing doctrine and unprecedented technology better than the competing principles advanced by petitioner.

Petitioners and some *amici* would have this Court believe that the Court of Appeals' references to "method of operation" are talismanic, over-literal, or insensitive to the nuances of distinguishing "expression" from "idea." None of this is so. "Expression" and "idea" are themselves imperfectly exclusive categories. As this Court has recognized since *Baker* v. *Selden*, 101 U.S. 99 (1879), expression may sometimes merge into idea. This intersection of the idea and expression is known to both the common law and the Copyright Act as "method of operation." The problem of method of operation as a fusion of idea and expression is sometimes seen outside the domain of computer software. Inside that domain, however, the issue is more complicated and far more significant than it usually is with other forms of activity covered by copyright. The Court of Appeals correctly understood the special functional role played by the menu command hierarchy in this record. On the basis of the District Court's detailed findings, the Court of Appeals reached the conclusion that best effectuates the broad Congressional intent lying behind §102(b). The alternative, which petitioner recommends without acknowledgment of the full consequences, leads to the conclusion that someone can copyright the use of any particular language as a means of communication with and between computers.

The Court of Appeals' approach also explains the distinction between the scope of copyright as previously

applied to computer programs, and the limitation implied by §102(b). The Court of Appeals denied copyright coverage to the language which petitioner's product, Lotus 1-2-3, understands, and in which users and other computer programs address it in order to acquire the results of its operation. It left intact the pre-existing degree of copyright coverage for the "source code," including non-literal copying of source code, or the form in which Lotus 1-2-3 is fixed in tangible media of expression. The Court of Appeals held only that the idea of a program which speaks the language of Lotus 1-2-3 is not covered by copyright, and that someone else may, without infringing petitioner's rights, create another computer program that speaks the same language. Despite all of the attempts by petitioner and others to make this conclusion appear stunning, confused or unprecedented, it is simple, and simply right.

Thus, contrary to the claims of petitioner and some *amici*, the Court of Appeals' approach imposes no change in copyright doctrine on any material deemed copyrightable before the opinions of the District Court below. This form of statutory construction, which leaves existing rights in place while refusing to infer broad statutory coverage over new technological and social developments, is an appropriate and restrained response to unprecedented problems unforeseen by the legislature. Congress never actually foresaw the technological issues that led to the need for statutory interpretation here. The approach taken by the Court of Appeals intelligently and sensitively applies the conceptions of the Act to a fundamentally altered technological context. Petitioner's proposed reading of the statute in its new context is not supported by an equivalently sophisticated statutory analysis, and leads to absurd and socially harmful results.

It was rightly rejected by the Court of Appeals. If Congress, upon informing itself and assessing the balance of social needs, wishes to achieve the results petitioner urges, it can do so. But this Court, in the quest for the legislature's elusive intent, need not accept petitioner's self-interested preference for the undeniably broad consequences of its position.

Argument

I. The Court of Appeals Correctly Applied §102(b) of the Copyright Act

Copyright law treats certain forms of expression as property and imposes legal penalties on practices which infringe, through copying or otherwise, the proprietary rights of the author or other owner. Congress has recognized that copyright coverage involves a tradeoff: it provides an incentive for some authors to produce work they might not otherwise undertake, but by withdrawing some forms of expression from free circulation, it may impair the production of other work that will also "promote the Progress of Science and useful Arts," U.S. Const., Art. I, sec. 8, cl. 8. That view is the source of the "most fundamental axiom of copyright law": that "[n]o author may copyright his ideas or the facts he narrates." *Feist Publications, Inc.* v. *Rural Tel. Service Co.*, 499 U.S. 340, 345 (1991) (quoting *Harper & Row, Publishers, Inc.* v. *Nation Enterprises*, 471 U.S. 539, 556 (1985)). Congress expressed that idea in §102(b) of the Copyright Act, which expressly provides that "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. §102(b). All parties agree that this formulation codifies the "idea-expression" distinction in the law of copyright. See *Feist*, 499 U.S., at 356; H.R. Rep. No. 94-1476, p. 57 (1976); S. Rep. No. 94-473, p. 54 (1975). The fundamental flaw in petitioner's argument is its failure to recognize that Lotus 1-2-3's menu hierarchy

and the Borland "key reader"—the two elements at issue in this case—lie at the intersection of these two ways of expressing the basic limits of copyright coverage: they are an amalgamation of idea and method of operation.

A. Critical Technical Distinctions Exist Between Computer Software and Other Literary Works

Petitioner downplays the significance of §102(b) by observing that every computer program is a "procedure, process, [or] system" whose source code specifies a "method of operation." Thus, it says, §102(b) cannot be read literally, or to exclude the menu hierarchy and macro language of Lotus 1-2-3 from copyright coverage, without eliminating all copyright coverage for computer programs. See, e.g., Pet. Br. at 28-29, 30-31, 43, 45.

But petitioner is simply wrong. As we show below, at the time Congress enacted the 1976 Act and the 1980 software amendments, it manifested a clear intention to cover specific types of computer-related expression. Congress did not, however, suspend the general structure of copyright law, which presumes that literary works can and do embody uncopyrightable ideas, systems, or methods of operation, as well as copyrightable expressions. Petitioner's conflation of copyrightable expression with non-copyrightable ideas and methods of operation ignores the unique nature of computer software among the class of literary works. Appropriate construction of the Copyright Act requires attention to this context. While some aspects of the Court of Appeals' opinion could be more precisely expressed, the decision below is far more sensitive to the distinctive

problems of computer programs as literary works than is petitioner's argument.

Terminology is important here. A program such as Lotus 1-2-3 is written in a computer language intended for use by specialized computer programmers. Tens of thousands of lines of instructions written in that primary language comprise the "source code" for Lotus 1-2-3. With appropriate processing by other computer programs, that "source code" is translated to "object code," a form of instruction suitable for execution by a particular type of computer.[1] Under the control of the appropriate object code, a computer can behave in ways intended by the programmer who created the source code instructions.

All parties agree that Congress in adopting the 1980 Amendments to the Copyright Act intended to effectuate the recommendation of the National Commission on New Technological Uses of Copyrighted Works that computer programs be treated as literary works. See Pub. L. No. 96-517, §10(a), 94 Stat. 3015, 3028 (1980). But as the previous legislative history of the Copyright Act reveals, Congress intended §102(b) of the Act to apply strictly in the technologically distinctive context of computer programs:

> Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or process adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the

1. These and other technical matters are more fully described in the Brief Amicus Curiae of Computer Scientists in Support of Respondent, at 4-6.

> expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law. H.R. Rep. No. 94-1476, p. 57 (1976).

Applied to the comparatively simple technological context in which Congress legislated in 1976, the intention was clear: what the programmer writes, the source code, is covered by copyright; what the program *does*, the idea or method of operation of the program, is not.

But, as Congress fully expected, the technological context changed substantially after 1976 and 1980. Programs of a type only familiar in the world of larger "mainframe" computers reached the mass consumer market along with the personal computer in the early 1980s. *See* Pet. Br. at 11 (Lotus 1-2-3 first released in 1983). Computer programs of the type and sophistication of Lotus 1-2-3 are *interactive*: they behave in complex ways following directions submitted by human users or other computer programs. Computers animated by such interactive programs, unlike all the other tools conceived by humankind in its long history of tool-making, respond to language. Put in terms of the Copyright Act, an interactive language, shared between Lotus 1-2-3 and a person or computer program directing the interaction, is the method of operating Lotus 1-2-3. This language is not the language of the source code, in which Lotus 1-2-3 was expressed by its creator, but another language whose purpose is the *operation* of the program.

Lotus 1-2-3 is an inherently complex program, which manages sophisticated mathematical and textual objects

called spreadsheets, draws charts and graphs, contains database storage and retrieval capacities, and so forth. The language in which a program such as Lotus 1-2-3 receives and executes directions from humans or other computer programs is not English, or like English. Usually two or three typewriter keystrokes (letters, numbers, or some punctuation) are an entire command, or a short series of such keystrokes initiates a multi-step process. For example, if a Lotus 1-2-3 user wants to direct her machine to save a worksheet, she might type the phrase "/WS" along with the name she wishes the saved worksheet to have. The command language of a program like Lotus 1-2-3 is quite large.[2] Like Esperanto, or any other invented language, the number of ideas that can be expressed in the Lotus 1-2-3 language is essentially unlimited, within the context of the spreadsheet program and the activities a user might desire it to perform.

The language of Lotus 1-2-3 is not spoken only by the program itself and its human users. Other programs, usually called "macros," can be written in Lotus 1-2-3 language that will allow Lotus 1-2-3 to perform operations under the macro's control. Just as a human user can type certain keys to form words in Lotus 1-2-3 language (in which case the program presents on the screen "menus" showing possible relevant choices as each word is formed) a macro program can be directed to control Lotus 1-2-3's operation, in which case—although the same words are presented—the program behaves

2. Petitioner refers in its brief to the 469 words contained in the menu hierarchy. Pet. Br. 5. This is not an exact enumeration of the number of commands in the language of Lotus 1-2-3, *see* Pet. Br. 8 n.12, but the magnitude is roughly comparable.

slightly differently. It need not show the menus while reading the macro, which is a document written in the Lotus 1-2-3 command language by the user herself, or by some other person.

The "menu hierarchy" included in Lotus 1-2-3 is essentially therefore akin to the words in a dictionary. It enables users who cannot remember the "vocabulary" (that is, the strokes on the keyboard) necessary to instruct Lotus 1-2-3 in particular circumstances to see relevant choices displayed on the screen. Respondent devised its Quattro Pro program to speak the same language as Lotus 1-2-3. Its menu hierarchy, if the user chose to enable the "emulation," reminded the user of relevant valid words in Lotus 1-2-3 language. This reminding function is a "user interface." Respondent's Quattro Pro key reader, *see Lotus Development Corp.* v. *Borland International, Inc.*, 831 F. Supp. 223, 226-27 (1993)(description of key reader), is another dictionary—one that "translates" instructions formed in Lotus 1-2-3 language into language that Quattro Pro recognizes, without display of menus. Its purpose was to allow "macros," that is, other computer programs that speak Lotus 1-2-3 language, to communicate with Quattro Pro. The central claim of this lawsuit, which the Court of Appeals properly rejected, is that the language recognized by Lotus 1-2-3 is copyrightable expression, rather than an uncopyrightable idea or method of operation under §102(b).

B. User Interfaces are Language Systems, or Methods of Operation, Excluded from Copyright by Section 102(b)

No one has ever copyrighted an entire language. Indeed, such an attempt would be absurd. Nor, by copyrighting a dictionary, can the dictionary's author withdraw the words contained within the dictionary from free discourse. But that, ultimately, is what petitioner seeks to do. Because the technological context is somewhat obscure, petitioner adduces arguments that would never be accepted in the context of other literary works, while at the same time contending that computer programs are literary works no different than all others for the purpose of §102(b).

The difference between computer programs and non-interactive literary works is significant, but not because it changes the nature of copyright law's application. Non-interactive literary works do not often create new languages; interactive ones, such as Lotus 1-2-3, often do. But the nature of petitioner's claim is made clear, as well as the overwhelming obstacles it faces, when another kind of language-inventing literary work is considered.

In 1939, for example, James Joyce published the renowned (if infrequently read) modernist novel called *Finnegans Wake*. Much of this novel was written in language of Joyce's own invention which, by a process which was in some respects the subject of the novel, English-speaking readers could partially understand. A sentence taken at random from the novel says:

> Cycloptically through the windowdisks and with eddying awes the round eyes of the rundreisers, back to back, buck to bucker, on their airish chaunting car, beheld with intouristing anterestedness the clad pursue the bare, the bare the green, the green the frore, the frore the cladagain, as their convoy wheeled encirculingly abound the gigantig's lifetree, our firleaved loverlucky blomsterbohm, phoenix in our wood-lessness, haughty, cacuminal, erubescent (repetition!) whose roots they be asches with lustres of peins. *Finnegans Wake* 55 (1976 ed.) (orig. publ. 1939).

Joyce's novel contains hundreds, indeed thousands, of unfamiliar words conjoined to make a language in which others could express their own ideas. There is no doubt that, when fixed in any tangible medium of expression, *Finnegans Wake* can be covered by copyright. But petitioner's argument demands that any literary work written using large numbers of new words copied from *Finnegans Wake* infringes the copyright. This argument confuses James Joyce's expression in his invented language, as fixed in the literary work called *Finnegans Wake*, and his idea: the invented *Wake*-language itself and the words comprising it. This is the very confusion that §102(b) of the Copyright Act is designed to prevent.

Respondent's Quattro Pro is an interactive literary work, a computer program, that understands the language of petitioner's Lotus 1-2-3. When expressions in this language are presented to it, whether a user is typing on a keyboard or a macro program is executing, Quattro Pro behaves responsively. A non-interactive literary work can also have this property with respect to the language of another. An author might prepare a dictionary of the

language of *Finnegans Wake*, for example, and for each unfamiliar word in that work give a definition based on sound, context, or other guides to meaning. Such a dictionary would "understand" the language of *Finnegans Wake*, and each word defined in it would have been copied from the novel. Petitioner must be prepared to argue that such a dictionary infringes the original copyright, and such an argument falls foul of §102(b).

This is true *a fortiori* of the literary works in question here. Petitioner claims that the Court of Appeals' reliance on the words "method of operation" is fallacious, and broad enough to deprive all computer programs of copyright protection. Pet. Br. 42-43. But the Court of Appeals saw correctly that the present case is governed by the clear language of the statute. Petitioner's claim to copyright coverage falls foul of §102(b) because it is seeking to bring within its copyright a system of ideas—the interaction language of Lotus 1-2-3. That language, as the Court of Appeals understood, is Lotus 1-2-3's "method of operation," as distinct from the language of Lotus 1-2-3's source code, which is the copyrightable expression of Lotus 1-2-3's authors. The phrase "method of operation," of the ones Congress used in §102(b), most closely describes the nature of the idea, as opposed to expression, petitioner seeks to cover. But, as petitioner agrees, the central question is not whether Lotus 1-2-3's command language is a "system," a "method of operation," or a "concept." If, as would be clear in the context of a non-interactive literary work, invented language is an "idea," it cannot be copyrighted.

In this context, it is easier to comprehend the relevance of the case of *Baker* v. *Selden*, 101 U.S. 99 (1879), to the present dispute. *Baker* concerned the

attempt to extend copyright to cover allegedly infringing grid-ruled forms resembling those published by the plaintiff, of a type devised to facilitate practicing a particular system of bookkeeping. The decision of this Court formalized the idea-expression distinction, holding that Selden could not cover by copyright the idea of his system of municipal bookkeeping. *Id.*, at 101.

This Court's decision in *Baker* v. *Selden* also gave rise to the so-called "merger doctrine," holding that expressive content is not covered by copyright when it is a "necessary incident" to an idea, which cannot be conveyed to another without some illustrative incident expression. *Id.*, at 103. The Court of Appeals correctly regarded Borland's use of a menu-like dictionary of Lotus 1-2-3's interaction language, intended to assist the user in forming the words of that language, as a necessary incident to that idea. This view is consistent with *Baker*, in which the Court found Selden's column headings on his ruled forms, such as "Balance Forward," "Adjusted," and "Additional Disbursements" to be excluded from coverage. These column headings, in the non-interactive context of Selden's forms, are like the items "Copy," "Recalc," and "Combine" found in the menu hierarchy of Lotus 1-2-3 and Quattro Pro.

This is so despite the fact that inventing a language requires originality, and is expressive. Petitioner points out that Lotus 1-2-3's creator, Mitchell Kapor, worked closely with others to invent a language that would "intelligently" facilitate communication with Lotus 1-2-3. Pet. Br. 9. Perhaps James Joyce worked alone in devising the language of *Finnegans Wake*; in both cases the expressive originality involved, however substantial, is not covered by copyright. As this Court said in *Feist*

Publications, Inc. v. *Rural Tel. Service Co.*, 499 U.S. 340 (1991), about the closely-related problem of factual compilation:

> It may seem unfair that much of the fruit of the [creator's] labor may be used by others without compensation. ... [T]his is not "some unforeseen byproduct of a statutory scheme." It is, rather, "the essence of copyright," and a constitutional requirement. The primary objective of copyright is not to reward the labor of authors, but "[t]o promote the Progress of Science and useful Arts." ... To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work. *Id.*, at 349-50 (citations omitted).

The approach taken by the Court of Appeals below coincides with the teaching of *Feist*. Its construction of §102(b) in the factual context of this case accords, as petitioner's does not, with the fundamental axioms and purposes of copyright.

II. Petitioner's Interpretation Leads to Absurd and Undesirable Results

Petitioner urges this Court to reject the Court of Appeals' approach, contending that if the Congressional intent to protect computer programs as literary works is to be effectuated, §102(b) cannot exclude from copyright coverage the command language of Lotus 1-2-3. As we have explained above, this contention is wrong. It depends on a confusion between the *source code* of Lotus

1-2-3, which is covered by copyright as an expression in a tangible medium of the method of operating a computer, and Lotus 1-2-3's language of interaction with users and other computer programs—its "user interface"—which is an invented language within the literary work and is thus an idea excluded from coverage by §102(b).

On the basis of this confusion, petitioner makes a claim never before made in the history of our copyright law. Petitioner asserts that *any* interactive literary work that understands the language of Lotus 1-2-3, behaving responsively and similarly when commands in that language are addressed to it, infringes the copyright coverage accorded Lotus 1-2-3's source code. This, petitioner asserts, is so despite the absence of any copying from the program as fixed in the tangible medium of expression required by the Act.

Petitioner's position is not only unprecedented, it leads to impossible consequences. Suppose Lotus 1-2-3's designers had elected, instead of devising their own invented language of interaction, to use English. In a technological context likely to be commonplace in the near future, in which personal computers can be directed by voice recognition equipment, such a decision would not be implausible. If Lotus 1-2-3 responded intelligently to every single English sentence that could be usefully spoken to a spreadsheet program (such as "Save this spreadsheet under the name Mary," "Move column 1 to column 5, please," and "Put zero in every box on row 12"), would any other spreadsheet program that recognized English sentences infringe? Would all other spreadsheet programs be compelled to offer an interface to users and other programs based around French, Urdu,

or Nahuatl, or else purchase a license to recognize English?[3]

This absurdity is the root of the problem with petitioner's contentions. The policy of the copyright system, as all concede, is to encourage the diffusion of useful knowledge. Petitioner's argument requires the conclusion that §102(b) effects that encouragement by requiring each computer program to speak a language different from all others, making it more difficult for both users and other computer programs to communicate effectively. "Compatibility," which is one of the central concerns of interactive software developers, is sacrificed to the desire of authors to recoup material rewards from the invention of new languages. This position conforms neither to the words of §102(b) nor to the Constitution's clear statement of the policy that enables the legislation. As this Court said in *Feist*, "the primary objective of copyright" is the encouragement to build upon information and ideas. 499 U.S., at 349-50. Petitioner's argument places much emphasis upon the incentive to authors provided by material rewards. Pet. Br. at 49. As we show below, concern that the Court of Appeals interfered with this legitimate interest is unfounded. What petitioner does not acknowledge, however, is that this interest is not the central policy of copyright. Extending copyright to cover all subsequent uses of

3. Petitioner might object that English, as opposed to the command language of Lotus 1-2-3, has been long in the public domain. No doubt this would be another valid reason for refusing to permit expansion of copyright to cover the use of English as a command language, but putting this point to one side, the example of existing natural languages helps to illustrate the way in which petitioner misunderstands the idea-expression distinction embodied in §102(b).

invented languages contained in a literary work will profusely reward some authors. But only at the expense of violating the fundamental axiom—of both our copyright system and our entire constitutional order—that ideas are free.

III. The Court of Appeals Properly Refused to Expand the Statutory Monopoly Without A Clear Statement of Congressional Intent to Alter the "Fundamental Axiom" of Copyright

As in *Sony Corp.* v. *Universal City Studios, Inc.*, 464 U.S. 417 (1984), this Court is asked to expand significantly the scope of the statutory copyright monopoly in a situation Congress did not fully foresee when the legislation was written. In *Sony*, the Court rejected an "unprecedented" attempt to seek relief against alleged infringement from the manufacturers of articles of commerce, themselves not covered by copyright, that could be used by consumers to make copies of copyrightable material. *Id.*, at 421. As in this case, the technological environment presented challenges unconsidered by the legislature. As this Court noted, "[i]n a case like this, in which Congress has not plainly marked our course, we must be circumspect in construing the scope of rights created by a legislative enactment which never contemplated such a calculus of interests." *Id.*, at 431.

The Court of Appeals was fully mindful of that responsibility here. Contrary to the contention of petitioner and some *amici*, the Court of Appeals' holding does nothing to contract the scope of copyright coverage

that Congress plainly intended in 1976 and 1980, when it accorded coverage to the expression of a computer program, its source code, and excluded its method of operation, or idea. Nor did the Court of Appeals fail to heed the Congressional reaffirmation, contained in the 1980 Software Amendments, that computer programs are literary works, to be so treated for the purposes of ascertaining copyrightability *vel non* under §102(b).

Expanding copyright coverage to include invented languages contained within a literary work is an enormous step, at odds with traditional copyright conceptions and the wording of §102(b). In the particular context of computer software, where the technological consequences of the decision could not have been before the minds of the legislators at the relevant time, such a decision is properly left to the Congress. As we have showed above, the method of operating an interactive computer program is to direct it, in its respective command language, to perform some task. If Congress wishes to alter §102(b), or undertake to cover computer software differently from other literary works, it can of course do so. As this Court said in *Sony*, "Congress has the constitutional authority and the institutional ability to accommodate fully the varied permutations of competing interests that are inevitably implicated by such new technology." 464 U.S., at 431. "Bluntly put," petitioner says, "Congress did not provide courts with a blank slate to write the rules of protection for computer software." Pet Br. at 49. Just so. For that reason, the Court of Appeals prudently declined petitioner's invitation to refashion the idea-expression distinction in the computer software context.

Conclusion

For the foregoing reasons, amicus urges that the decision of the Court of Appeals be affirmed.

Respectfully submitted,

EBEN MOGLEN
Counsel of Record
Columbia Law School
435 West 116th Street
New York, NY 10027
212-854-8382/7946 (Fax)

PAMELA S. KARLAN
580 Massie Road
Charlottesville, VA 22903
804-924-7810/7536 (Fax)

Counsel for Amicus Curiae

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

No. 94-2003

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL INC.,
Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

Brief *Amicus Curiae* of Software Forum In Support of Respondent

DIANE MARIE O'MALLEY
Counsel of Record
MABEL NG
HANSON, BRIDGETT, MARCUS, VLAHOS & RUDY
333 Market Street, Suite 2300
San Francisco, CA 94105-2173
(415) 777-3200
Attorneys for Amicus Curiae SOFTWARE FORUM

December 1995

BEST AVAILABLE COPY

28 pp

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
INTEREST OF AMICUS CURIAE	1
SUMMARY OF ARGUMENT	4
ARGUMENT	6
I.	
THE NEED FOR A "BRIGHT LINE" TEST	6
A. The Absence Of A Bright-Line Test Deters Software Innovation	6
B. "Fair Use" Analysis Would Be Undesirable As A Basis For Decision	8
C. "Literal/Non-literal" Analysis Is Similarly Problematic	10
D. Extending Copyright Protection To Functional Elements Such As Menu Command Hierarchies Could Undermine The Software Industry As A Whole	11
II.	
MENU COMMAND HIERARCHIES ARE FUNCTIONAL AND HENCE NOT COPYRIGHTABLE	16
A. Even Standing Alone, The Command Hierarchy Is An Unprotected "System"	18
B. The Menu Command Hierarchy Is An Unprotected Method Of Operation	19
CONCLUSION	23

TABLE OF AUTHORITIES

CASES

	Page
Baker v. Selden, 101 U.S. 99, 25 L.Ed. 841 (1879)	19
Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 109 S.Ct. 971, 103 L.Ed.2d. 118 (1989)	8n
Campbell v. Acuff-Rose Music, Inc., ___ U.S.___, 114 S.Ct. 1164, 127 L.Ed.2d 500 (1994)	9
Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992)	10
Digital Equipment Corp. v. C. Itoh & Co., Ltd., 229 U.S.P.Q. 598 (D.N.J. 1985)	14n
Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991)	8
Fogerty v. Fantasy, Inc., ___U.S.___, 114 S. Ct. 1023, 127 L.Ed.2d 455 (1994)	6
Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823 (10th Cir. 1993)	14, 14n
Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 105 S. Ct. 2218, 85 L.Ed.2d 588 (1985)	8
Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F.Supp. 78 (D. Mass. 1992)	3n, 7, 10n
Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F.Supp. 203 (D. Mass. 1992)	3n
Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F.Supp. 202 (D. Mass. 1993)	3n
Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F.Supp. 223 (D. Mass. 1993)	3n, 10n

TABLE OF AUTHORITIES

CASES

	Page
Lotus Development Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995)	*Passim*
Lotus Dev. Corp. v. Paperback Software Int'l, 740 F.Supp. 37 (D. Mass. 1990)	2, 3, 11
NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989)	12n
Sony Corp. of America v. Universal City Studios, Inc., 464 U.S. 417, 104 S.Ct. 774, 78 L.Ed.2d 574 (1984)	9
Xerox Corp. v. Apple Computer, Inc., 734 F.Supp. 1542 (N.D. Cal. 1990)	14

OTHER AUTHORITIES

P. Barnard, et al., *Learning and Remembering Interactive Commands,* 1982 Proc.: Hum. Factors in Computer Sys. 2	20n
K. Bury, et al., *Windowing vs. Scrolling on a Visual Display Terminal,* 1982 Proc.: Hum. Factors in Computer Sys. 41	20n
Computers & Intellectual Property: Hearings Before the Subcomm. on Courts, Intellectual Property, and the Admin. of Justice of the House Comm. on the Judiciary, 101st Cong., 2d Sess. 243 (1990)	4
Final Report of the National Commission on New Technological Uses of Copyright Works (1978)	15

TABLE OF AUTHORITIES

OTHER AUTHORITIES

	Page
A. Hauptmann & B. Green, *A Comparison of Command, Menu-selection, and Natural Language Computer Programs,* 2 Behav. & Info. Tech. 163 (1983)	20n
D.S. Karjala, *"Copyright, Computer Software, and the New Protectionism,"* Jurismetrics J. 33, 59-60 (Fall 1987)	13n
G.K. Landy, "Macros and Broad Software Copyright Protection: The Latest Round in Lotus v. Borland," *The Computer Lawyer,* Vol. 11, No. 2 at 25 (Feb. 1994)	15n
J. Rosenberg, *Evaluating the Suggestiveness of Command Names,* 1982 Proc.: Hum. Factors in Computer Sys. 12	20n
P. Samuelson & R.J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits,* 30 Jurismetrics J. 121 (Fall 1989)	17n
R. Stallman & S. Garfinkel, *Viewpoint: Against User Interface Copyright,* Comm. ACM, Nov. 1990, at 15	19n
R.M. Whitmeyer, Comment, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software,* 85 Nw. U. L. Rev. 1103 (1991)	8n

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL INC.,
Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

BRIEF *AMICUS CURIAE*
SOFTWARE FORUM

INTEREST OF THE AMICUS CURIAE

Software Forum (the "Forum") respectfully submits this brief *amicus curiae* in support of affirmance of the decision of the United States Court of Appeals for the First Circuit below, *Lotus Development Corp. v. Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir. 1995). The consent of the attorneys for both Respondent and Petitioner has been obtained.[1]

[1] Neither Lotus nor Borland is represented on the Forum's 20-member Board of Directors or its 15-member Advisory Board. The Forum, previously known as the Software Entrepreneurs' Forum, has filed *amicus* briefs in this matter with both the District Court and the Court of Appeals. This Brief was approved pursuant to the balloting procedures

The Forum, founded in 1983 and based in California's Silicon Valley, is a non-profit organization with a membership of more than 1,000 independent software developers, consultants and software providers. While the Forum is reluctant to burden this Court with another brief in this case, it believes it brings a valuable perspective to the issues pending before the Court. The Forum is the largest non-profit group in the United States that provides software entrepreneurs with resources needed to develop, market and sell software products. Its counsel are familiar with the briefs and questions presented. Its members are united by a desire to preserve their freedom to develop software products and businesses based on free-market principles and market-driven forces. Many of it members are precisely the "small developers" that the District Court and Lotus contended would be the "biggest losers of all" if Lotus's menu command hierarchy were not protected by copyright. *Lotus Dev. Corp. v. Paperback Software Int'l,* 740 F.Supp. 37, 75 (D. Mass. 1990) (*"Paperback"*), Pet. App. 183a, 249a. The Forum could not disagree more with the District Court's decision. Far from encouraging the creation of new software programs, the Forum views the position urged by Lotus and adopted by the District Court as leveling a potentially lethal blow to innovation and development in the software industry.

The Forum's members believe that there is additional argument needed on some important points which have not received sufficient attention. First, the Forum wishes to emphasize that the extension of copyright protection to computer menu command hierarchies inhibits the efforts of all — consumer and developer — to work in an organized environment. The broad copyright granted Lotus by the District Court frustrates this country's ability to compete

---

of the Forum which exercised complete control over its editorial content. Borland helped to defray the cost of printing.

and excel by foreclosing the use of standard items for the ultimate benefit and ease of use of all consumers (as was the case with Selden's System of double-entry booking which the Court of Appeals recognized as the now-almost universal T-accounts system). As aptly pointed out by the Court of Appeals, "under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used." *Lotus Development Corp. v. Borland Int'l, Inc.,* 49 F.3d 807, 817-818 (1st Cir. 1995) ("Lotus"), Pet. App. 20a. The Forum does not believe that such relearning benefits the ultimate user or the industry as a whole and believes that it runs contrary to the purpose of the copyright laws to enrich the public through access to creative works.

Second, the freedom to develop software products is jeopardized if this Court fails to affirm the decision of the Court of Appeals. Indeed, the disturbing implications of the decisions of the District Court[2] and its reliance on *Paperback* so troubled the Forum's members that the Forum filed *amicus* briefs on the merits both in the District Court and the Court of Appeals. It does so here again with the hope of persuading this Court to preserve a fundamental characteristic of the software industry's innovation and development.

Ironically, in testimony before Congress immediately before this lawsuit was filed, Mitchell Kapor, the principal

---

[2] *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 788 F.Supp. 78 (D. Mass. 1992) (*"Borland I"*), Appendix to the Petition for Certiorari ("Pet. App.") 145a; *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 799 F.Supp. 203 (D. Mass. 1992) (*"Borland II"*), Pet. App. 106a; *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 831 F.Supp. 202 (D. Mass. 1993) (*"Borland III"*), Pet. App. 71a; and *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 831 F.Supp. 223 (D. Mass. 1993) (*"Borland IV"*), Pet. App. 29a.

developer of Lotus 1-2-3, captured what is at stake in this case:

> If you want to keep this industry as vibrant and successful as it's been, then a properly constructed intellectual property policy will respect protection but give preference to innovation. Over-protection of intellectual property is as pernicious as under-protection in its stifling effects on innovation and consequent loss to society.

Computers & Intellectual Property: Hearings Before the Subcomm. on Courts, Intellectual Property, and the Admin. of Justice of the House Comm. on the Judiciary, 101st Cong., 2d Sess. 243 (1990). To allow copyright protection of purely functional elements such as the 1-2-3 menu command hierarchy will have precisely such a "pernicious" effect on "innovation and consequent loss to society." The Forum will elaborate upon these points as succinctly as possible in the pages that follow.

SUMMARY OF ARGUMENT

This *amicus* brief makes two points. First, the Forum believes that in order to foster continuing innovation in the software industry, the scope of copyright protection of computer programs must be reasonably ascertainable and predictable. The uncertainty created by an *ad hoc* test necessarily deters new software development. The wider the scope of uncertainty, the greater the chilling effect. The resort to two modes of analysis in the proceedings below — the "fair use" doctrine discussed (but not endorsed) by Judge Boudin and the "literal/non-literal" analysis used by the District Court — necessarily leaves software developers with no means of assessing whether their conduct is permissible under the copyright laws.

Second, the Forum urges the Court to find menu command hierarchies to be uncopyrightable subject matter because of the value to public access obtained by allowing the most efficient means of performing a task to be available. Menu command hierarchies are not "computer programs" or "screen displays." Rather, they are an output of a program that in turn may be operated by the user to control the program. As the Court of Appeals correctly analogized, it is as though a video cassette recorder ("VCR") generated the "virtual" buttons that the user then employs to operate the VCR and view its screen. These VCR virtual buttons are not a proper subject of copyright.

The command hierarchy does not have an existence separate from the control function. A menu command hierarchy is the method by which a user operates a computer, just as surely as a handle is the method by which a user operates a hammer and a telephone keypad is the method by which a user operates a telephone. The Forum urges the Court to adopt a bright-line rule holding that such functional elements are unprotected under the copyright laws.

The Forum believes that the need for such clear demarcation is especially pressing in the software industry and that neither "fair use" nor "literal/non-literal" analysis provides the Forum members or other developers with a framework for making a reasonable evaluation of the scope of protection for software under the copyright laws. The District Court's decision also does not recognize the importance of the ultimate user's investment in learning the menu command hierarchy and what would be lost if the user had to learn a difficult menu command over and over and over each time he or she used a new program.

ARGUMENT

I.

THE NEED FOR A "BRIGHT LINE" TEST

This Court has recognized that the monopoly rights that Congress granted under the Copyright Act are limited in nature and have as their purpose "enriching the general public through access to creative works." *Fogerty v. Fantasy, Inc.*, ____ U.S. ____, 114 S. Ct. 1023, 1030, 127 L.Ed.2d 455, 465 (1994). To serve that purpose, this Court has held that "it is peculiarly important that the boundaries of copyright law be demarcated as clearly as possible." *Id.*

A. The Absence Of A Bright-Line Test Deters Software Innovation.

In the Forum's view, the need for clear rules identifying the boundaries of copyright protection, while always acute, is rendered even more so by three characteristics of the software industry. First, many if not most software developers are small entrepreneurs, not large corporations. These small firms have provided much of the force behind the successive waves of innovation that have propelled forward the information revolution. Such firms are extremely susceptible to the chilling effects of uncertain legal rules. Small entrepreneurs simply do not have the resources to support years of litigation in geographically-remote courts in order to get a product to the market. And even were such resources available, the cloud placed over a new product by lengthy litigation is potentially lethal not only to the product but also to the small developers' very existence.

In this respect, Borland's fate has provided developers with a cautionary note that has chilled the industry. When the District Court rendered its initial decision imposing an injunction on Borland, the developer community was incredulous. By the time Borland, which by the standards of most

small developers is a *large* company, was able to get the injunction lifted, the company was driven out of the spreadsheet market. For Forum members, as well as for the industry, there is only one message — that even "unsuccessful" copyright litigation can crush smaller competitors. If the scope of copyright protection is in any way uncertain, prudent small software developers will simply avoid risks and curtail any new development.

A second characteristic of the software industry that adds to the need for bright-line rules is the short lifespan of most software products. Products placed under the cloud of litigation likely will be obsolete or unwanted by the time the product's developer ultimately prevails in litigation. Borland's experience again is illustrative. Two years after litigation began, Lotus still had not formulated "which elements of 1-2-3, separately or in combination, were copyrightable," much less which were copied. *Borland I*, 788 F.Supp. at 98, Pet. App. 180a. Unlike patent cases, where the entire prosecution and claim process is designed to put competitors and others on notice at the outset of the bounds of the claimed invention, a developer sued in a copyright case might still be guessing as to what was infringed more than half way through the life of a software program with a four-year life cycle. To steer clear of such traps, developers will avoid committing substantial resources to the development of new products if there is any uncertainty at all regarding the scope of copyright protection of some other software product.

Finally, the software industry needs clear guidelines because copyright issues in the field are likely to be recurring. Like most other technologies, software engineering advances incrementally. Software developers rarely construct programs that perform basic functions from scratch. Instead, they resort to software libraries or to the technical literature

for efficient algorithms or modules of computer code.[3] In other words, software developers "build freely upon the ideas and information conveyed" by other programs. *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 349-350, 111 S.Ct. 1282, 1289-1290, 113 L.Ed.2d 358 (1991). Uncertainty regarding the extent to which such incremental advances are permitted under copyright law strikes at the very heart of the process by which software innovation takes place.[4]

B. "Fair Use" Analysis Would Be Undesirable As A Basis For Decision.

In his concurring opinion in the Court of Appeals, Judge Boudin observed that a different approach for decision in this case is an adaptation of the "fair use" doctrine. *Lotus,* 49 F.3d 807, 821; Pet. App. 27a-28a (Boudin, J., concurring). As Judge Boudin recognized, however, application of the fair use doctrine "would cause cost and delay," and "would also reduce the ability of the industry to predict outcomes." *Id.* at 821-822; Pet. App. 28a. Indeed, fair use requires *ad hoc* determinations that are the antitheses of the kind of bright-line rule that the software industry urgently needs.

The indeterminate nature of fair use analysis has been made clear by this Court. For example, in *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 560, 105 S. Ct. 2218, 2230, 85 L.Ed.2d 588, 607 (1985), this Court said that because the doctrine is an equitable rule of reason, "no

---

[3] *See, e.g.*, R.M. Whitmeyer, Comment, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software,* 85 Nw. U. L. Rev. 1103, 1110 (1991).

[4] The Court's patent cases recognize this incremental process as lying at the heart of technological innovation. *See, e.g., Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 156-157, 109 S.Ct. 971, 980-981, 103 L.Ed.2d. 118, 137-138 (1989).

generally applicable definition is possible, and each case raising the question must be decided on its own facts." *Accord Sony Corp. of America v. Universal City Studios, Inc.*, 464 U.S. 417, 448 n. 31, 104 S.Ct. 774, 792 n. 31, 78 L.Ed.2d 574, 596 n. 31 (1984). Similarly, in *Campbell v. Acuff-Rose Music, Inc.*, ____ U.S. ____, 114 S.Ct. 1164, 1170-1171, 127 L.Ed.2d 500, 514-515 (1994), this Court emphasized that fair use "calls for case-by-case analysis" and that the four statutory factors listed in the Copyright Act are not to be treated in isolation, but rather "[a]ll are to be explored, and the results weighed together, in light of the purposes of copyright."

Consideration of the fair use factors serves to highlight further the lack of correspondence between the issues in this case and the concerns to which fair use is directed. For example, under the first factor, Judge Boudin recognized that Borland's commercial use could weigh against a finding of fair use. *Lotus,* 49 F.3d at 821, Pet. App. 27a-28a. Yet, the commercial nature of Borland's use in which Borland was "merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros" is at the heart of the conduct that Judge Boudin thought should be protected. *Id.*, Pet. App. 27a. Similarly, the fourth factor weighs against a finding of fair use if the defendant's conduct "would result in a substantially adverse impact on the potential market" for the original. *Campbell,* 114 S.Ct. at 1177. It is precisely in such instances of potential market impact, however, that Judge Boudin believed that the defendant's conduct should be privileged. *Lotus,* 49 F.3d at 822, Pet. App. 28a. ("Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.").

Attempting to resolve this case by adapting fair use analysis is thus an effort to force a square peg into a round hole. The effort is likely to lead to even greater uncertainty

than is already inherent in fair use analysis. For software developers, therefore, fair use analysis is a singularly ill-suited vehicle for addressing the issues posed by this case.

C. "Literal/Non-literal" Analysis Is Similarly Problematic.

In expressly eschewing the application of bright-line rules,[5] the District Court took the position that the organization, structure and labels attached to the 1-2-3 menu hierarchy were "non-literal" elements of a computer program and that these "non-literal" elements, together with a program's "literal" elements, are copyrightable.[6] As developers, the Forum is concerned about its members' ability to predict whether a particular element will be protected under this literal/non-literal analysis. In this respect, at least, the Forum concurs with the (somewhat understated) assessment of the Court of Appeals in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992), that "[t]o be frank, the exact contours of copyright protection for non-literal program structure are not completely clear."

The District Court's decision amply illustrates the *ad hoc* and unpredictable nature of determinations made under a literal/non-literal analysis. For example, the District Court concluded that Borland was a copier, at best, and a pirate, at worst, despite the facts that (1) Borland was not found to have copied, paraphrased, translated or used any of Lotus's code; (2) Lotus did not contend (and, so far as the Forum has been able to ascertain, indeed has tacitly conceded) that Borland's code did not use the same structure or organization as Lotus's code; and (3) Borland's product "looks substantially different" in terms of screen display, and hence did not infringe 1-2-3's pictorial displays. Borland, in fact,

[5] *Borland I*, 788 F.Supp. at 92, Pet. App. 169a.

[6] *Borland IV*, 831 F.Supp. at 232-233, Pet. App. 43a-44a.

took far less of 1-2-3, from both a qualitative and quantitative point of view, than 1-2-3 took from such predecessor products as VisiCalc, VisiTrend/Plot, Multiplan, and Prokey. Yet, the District Court here, applying the literal/non-literal test, concluded that Borland's use of Lotus's command words was actionable, while it found in the earlier *Paperback* decision that Lotus's conduct in taking the spreadsheet metaphor, the use and identification of various function keys, the two-line moving cursor, the use of highlighting, and even the VisiTrend Plot command words themselves, was not only permissible but even worthy of approbation. *See Paperback*, 740 F.Supp. at 63-67, Pet. App. 227a-234a.

From the perspective of developers, these respective outcomes are frankly inexplicable and further complicate the already confusing copyright issues facing small developers. The *reason* the District Court reached such inexplicable outcomes, however, is easy to understand: as the Court of Appeals pointed out, the District Court's test asked the wrong question. *Lotus*, 49 F.3d at 815, Pet. App. 14a-15a. The test applied by the District Court in *Paperback* sought to distinguish between "the idea and its expression," *Paperback*, 740 F.Supp. at 65, Pet. App. 229a; but that is not the appropriate question to ask in determining the copyrightability of a functional work. To decide whether a wall socket is copyrightable, the question is not whether there is more than one way to express the "idea" of a wall socket or whether someone has used someone else's "expression" of the concept of a wall socket. Instead, with wall sockets — as with menu command hierarchies — the appropriate question is the one posed by the Court of Appeals: is the item functional and therefore foreclosed from protection by Section 102?

D. Extending Copyright Protection To Functional Elements Such As Menu Command Hierarchies

Could Undermine The Software Industry As A Whole.

It is significant that no software developers have expressed support for the position urged by Lotus. Indeed, support for the copyright scheme proposed by Lotus comes from an industry which has in past litigation unsuccessfully attempted to use copyright to prevent competition in the marketplace. Had the industry been successful in that litigation, it would have gained a monopoly position otherwise only available under patent law. Furthermore, that monopoly position would have lasted 75 years rather than the significantly shorter term available for patents. In addition, the safeguards in effect under patent law, specifically the requirements to disclose the invention, to present specific claims, and to show non-obviousness and novelty, would be completely evaded.

Illustrative is Intel's copyright litigation against rival NEC (and, three years later, against AMD). Intel long has enjoyed market dominance as the maker of the chips at the heart of all IBM-compatible personal computers. NEC, however, engaged in a full-scale effort to develop a microprocessor that would be compatible with Intel's but that would contain improved performance. Intel sued for copyright infringement on the ground that NEC had copied its "microcode," which is the set of roughly ninety machine-language instructions that tell the microprocessor which transistors to actuate. Ultimately, Intel's suit was unsuccessful as the District Court in that case found no infringement.[7] As one commentator summarized, what was at stake in the litigation, and in Intel's subsequent suit against AMD, was

[7] *See NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989).

its ability to monopolize the microprocessor market indefinitely:

> Given that writing the actual microcode is a relatively small part of the cost of overall chip development, it seems almost incredible that a company in [NEC's] position would copy microcode after independently developing everything else, unless they thought copying was necessary for compatibility or was dictated by the hardware environment. Even assuming copying, however, to find copyright infringement is to give to this most fundamental form of computer software technology a level of copyright protection never before afforded to a technological product. . . . [T]he defendant can wholly enjoin the sale of the plaintiff's chip, thereby depriving the public of the advantages of a technologically superior product. Moreover, this position will be retained for seventy-five years, or at least until the original chip becomes obsolete through later technological developments. This seems a very high price for the public to pay to remedy a "wrong" that would not be at all wrong or unfair if it involved any type of technology other than one that effects machine operation through manipulation of human-understandable symbols.[8]

Xerox has been no less ambitious in its effort to use the copyright laws. In the late 1970's, Xerox, the inventor of the "mouse," developed a mouse-driven computer that "alleg-

[8] D.S. Karjala, *"Copyright, Computer Software, and the New Protectionism,"* Jurismetrics J., at 59-60 (Fall 1987).

edly was the first to introduce fanciful visual displays and graphical images to aid user interaction with the computer." *Xerox Corp. v. Apple Computer, Inc.*, 734 F.Supp. 1542, 1544 (N.D. Cal. 1990). Xerox sued Apple, claiming that the copyright registration of Xerox's workstation included its computer program and screen displays. Although the suit was unsuccessful, Xerox had been remarkably forthcoming about the breadth of the monopoly that it sought. According to Xerox, by virtue of its copyright registration, it had "the right to control the future of graphic user interfaces in the 1990's and beyond."[9] *Id.* at 1545. Graphical user interfaces now are central to much of personal computing. Such a monopoly by Xerox, and the resulting lack of competition, would have significantly hindered the further development and refinement of that extremely important advance in the usability of software.

DEC and Gates Rubber each have sought specifically to use copyright protection of menu displays in litigation against their hardware competitors.[10] The example of *Gates Rubber* is particularly notable inasmuch as it shows that Lotus's proposed expansion of the copyright laws would be likely to have consequences well beyond the computer industry. Gates Rubber is the dominant manufacturer of rubber belts for use in industrial machinery and has developed a software program for performing the complicated calculations needed to determine the proper belt for a particular machine. It sued its competitor on the ground that the competitor's introduction of a rival software program violated Gates's copyright in a variety of functional

[9]The District Court responded to this assertion with the observation that "courts may on occasion break up monopolies, but should not create them." *Id.* at 1545, n.5.

[10]*See Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9 F.3d 823, 843-844 (10th Cir. 1993); *Digital Equipment Corp. v. C. Itoh & Co., Ltd.*, 229 U.S.P.Q. 598, 603 (D.N.J. 1985).

program outputs, including its menus and sorting criteria for the entry of data. *Gates Rubber*, 9 F.3d at 843-844. As the use of software spreads even to such typically industrial uses, it seems clear that Lotus's claim, described by one commentator as a "broadening of copyright protection for program command protocols without logical limit,"[11] ultimately would touch virtually every aspect of American business.

Finally, and closer to home, it is not only other hardware makers that likely would be the targets of such copyright claims. In its Final Report, CONTU noted that the emergence of a separate software industry was tied to the inability of hardware manufacturers to control entry into the software market. Citing to a study commissioned by it, CONTU observed:

> Whatever their historical dominance, the hardware corporations lack the ability to control entry into the software market, and . . . their market shares are being steadily eroded by the independents. Thus, we can tentatively conclude that protection of software . . . serves to benefit consumers by enhancing competition and increasing long-run supply.

M. Breslow, A. Ferguson and L. Haverkamp, *An Analysis of Computer and Photocopying Issues From the Point of View of the General Public and the Ultimate Consumer* at IV-13 (1978), *quoted in Final Report of the National Commission on New Technological Uses of Copyright Works* at 24 (1978).

This inability to "control entry," however, assumes that hardware manufacturers cannot obtain a copyright in the

[11]G.K. Landy, "Macros and Broad Software Copyright Protection: The Latest Round in Lotus v. Borland," *The Computer Lawyer*, Vol. 11, No. 2, at 25 (Feb. 1994).

specifications needed to operate their machines. Yet, that is *precisely* what Lotus seeks to claim. Lotus 1-2-3 serves as a platform for the operation of macros in exactly the same way that computer hardware serves as a platform for the operation of applications software. Obviously, Lotus's parent, IBM, and other computer makers would benefit enormously from the ability to monopolize the control-point to their machines and, accordingly, recapture their "historical dominance" in the production of software in this country. The losers, on the other hand, would be not only the software industry, but also consumers and the American economy as a whole.

II.

MENU COMMAND HIERARCHIES ARE FUNCTIONAL AND HENCE NOT COPYRIGHTABLE.

As the authors of thousands of software programs, the Forum members support copyright protection of computer programs. It is fitting that copyright law protects against a "software pirate's" unauthorized copying of diskettes, prohibits users from unauthorized copying, and prevents copying of source code directly or indirectly by translating that code into another programming language. Such protection against the copying of source code and object code is appropriate. Moreover, because the statute protects audiovisual and pictorial works, it is also appropriate that the screen displays generated by a computer program are protected.

A menu command hierarchy, however, is neither a pictorial work nor a computer program. Although the way that the command hierarchy is *displayed* conceivably could be found to be sufficiently expressive to warrant protection, Lotus has made clear in its brief that it does not claim that Borland copied "the color, style, layout, or format in which the words or menus are displayed on the screens." Brief for the Petitioner, p. 6.

At the same time — and a point that Lotus fails to address in a meaningful way — the menu command hierarchy is not protectable as a "computer program." The menu command hierarchy is an *output* of the program, not the program itself. In defining "computer program," Section 101 provides that a program "is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Pet. App. 272a. Although the menu command hierarchy may be used to generate such a "set of statements or instructions" — a kind of program known as a "macro" — it does not provide such an operational role itself. Put differently, it is the keyboard, not the composition.

A software developer can duplicate a menu command hierarchy quite closely without access to any of its competitor's underlying computer instructions, and Lotus apparently concedes that Borland did so here. Indeed, the Lotus code was never even entered into evidence in this case. Trial Tr. of Apr. 1, 1993, p. 5-136 to 5-137, Joint Appendix at 300. Consequently, this is a case where the copying of computer code or the structure, sequence or organization of the program is not at issue.[12] Instead, at issue is whether a program output is protected apart from the protection provided to pictorial and audiovisual works. The Court of

---

[12] Inasmuch as legal protection for computer programs is directed principally at providing an incentive for the creation of new programs, it is noteworthy that at least one survey has found the developer community to be hostile to extending intellectual property protection even to the structure or organization of computer code. According to the survey, an overwhelming majority supported intellectual property protection for source code (93%) and object code (85%), but they "strongly rejected the idea of protecting the 'structure, sequence, and organization' of computer programs." P. Samuelson & R.J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurismetrics J. 121, 129, 135 (Fall 1989).

Appeals correctly determined that it is not. Considered solely as output, the command hierarchy is an uncopyrightable "system." In context, however, there is a further reason why the command hierarchy is not protected: as the Court of Appeals found, the command hierarchy provides the program's "method of operation." Its role is no different than that played by pedals on a bicycle; and it is no more subject to protection under the copyright laws than are bicycle pedals.

A. Even Standing Alone, The Command Hierarchy Is An Unprotected "System."

Even considered apart from its functional role in operating a computer program, a menu command hierarchy is a non-copyrightable "system" under 17 U.S.C. Section 102(b). To illustrate, imagine a young Thomas Jefferson writing a program that creates a "virtual" Library of Congress. In running Mr. Jefferson's program, the user initially faces a screen depiction of the Library of Congress, which one can enter by clicking on the Library's front door. The user then sees a screen showing a large rotunda, with three wings marked "[ ]," "[ ]," and "Memory." Clicking on "Memory" calls up a screen showing a hallway divided between "Natural History" and "Human History." Clicking on "Human History" in turn reveals rooms labeled by country, and clicking the door for "U.S. History" reveals a room with many stacks. The stack marked "Civil War" has a shelf designated "Abraham Lincoln," and that shelf has on it the Gettysburg Address.

In considering the scope of copyright protection for this work, one can readily identify the source code and object code written by Mr. Jefferson as a protected computer program. The screen displays are pictorial or audiovisual works, and hence are protected as well. The Gettysburg Address, although generated in this instance through operation of the computer program, is in the public domain, and

hence is no more subject to copyright protection than are facts or ideas. Finally, the hierarchy by which the "books" are organized would not appear to be copyrightable, any more than such a system would be if devised wholly apart from the use of computers. To hold otherwise would seem to open up a radically expanded view of the scope of copyright, by which everything from the Dewey Decimal System to a system for arranging bookkeeping columns and headings could be subject to copyright — a result fundamentally at odds with *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879).

As the Court of Appeals recognized, the menu command hierarchy is also unprotected for another reason. The menu command hierarchy is not simply a system, but the means by which users can operate their computer program. This final point is addressed in the next section.

B. The Menu Command Hierarchy Is An Unprotected Method Of Operation.

As noted above, most if not all machines have what may be termed a "user interface." A user interface "is what the user must learn in order to operate a machine. The user interface of a typewriter is the layout of the keys. The user interface of a car includes a steering wheel for turning, pedals to accelerate and reduce speed, a lever to signal turns, etc."[13] In the case of a computer program, the user interface typically consists of the display on a screen of textual or graphic elements (icons) that enable the user to direct the operation of the program.

Computer programs have become increasingly universal, and that trend will only continue. For example, keyboards on computers, numeric pads on ATM machines and tele-

---

[13] R. Stallman & S. Garfinkel, *Viewpoint: Against User Interface Copyright*, Comm. ACM, Nov. 1990, at 15.

phones, and buttons on VCRs no longer operate mechanically, but instead operate through the use of software. Moreover, it seems likely that vocal commands will soon replace screen displays in many contexts. In such instances, there would be no visual depiction of the command hierarchy and no words attached to the hierarchy other than those spoken by the users themselves.

In all of these examples, what developers strive to maximize to the benefit of the ultimate consumer is not the "expressiveness" of the user interface, but its functionality. Indeed, an entire field of study known as "human factor analysis" is devoted to the empirical analysis of interface issues for purposes of optimizing efficiency and effectiveness. Thus, for example, researchers have examined choices among various abbreviations and command names,[14] the relative effectiveness of command-based, menu-driven, and natural language programs,[15] and the choice of windowing or scrolling.[16]

In his testimony in this case, Mitchell Kapor confirmed that he (like other developers) strived to devise a user interface with maximum efficiency and effectiveness. Thus, for example, menu command words were chosen to "intelligently convey to the user the purpose of each command and its underlying functionality." Affidavit of Mitchell D. Kapor ¶77, Pet. App. 291a. Kapor sought to achieve an organiza-

[14] *See, e.g.*, J. Rosenberg, *Evaluating the Suggestiveness of Command Names*, 1982 Proc.: Hum. Factors in Computer Sys. 12; P. Barnard, et al., *Learning and Remembering Interactive Commands*, 1982 Proc.: Hum. Factors in Computer Sys. 2.

[15] *See, e.g.*, A. Hauptmann & B. Green, *A Comparison of Command, Menu-selection, and Natural Language Computer Programs*, 2 Behav. & Info. Tech. 163 (1983).

[16] *See, e.g.*, K. Bury, et al., *Windowing vs. Scrolling on a Visual Display Terminal*, 1982 Proc.: Hum. Factors in Computer Sys. 41.

tion that would "reflect a structured approach that communicated the product's underlying functionality." In other words, Kapor followed sound principles of product engineering in constructing the user interface.

Certainly a well-engineered user interface entails considerable creativity and inventiveness. That is far different than the District Court's apparent holding that, especially in a feature of industrial design as to which there is theoretically an optimally efficient answer, one developer can bar others from building upon its efforts — without ever having shown that its own effort constituted a material advancement over the previous efforts of others. Lotus has never shown that its menu command hierarchy constituted a new and nonobvious improvement in the development of user interfaces. As the Court of Appeals found, Borland therefore should be free to build upon it. The Court explained, in reasoning with which the Forum concurs entirely:

> [I]n most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression. In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Lotus, 49 F.3d at 818, Pet. App. 21a.

If this Court were to reverse the Court of Appeals, it will substantially frustrate the efforts of the software industry to create software for the ultimate consumer in an organized and efficient manner. The Forum urges the Court to adopt a bright-line rule creating a "safe harbor" in which software developers are free to utilize the functional elements of the output of computer programs, including the program's menu command hierarchy. Such a rule will provide developers like the members of the Forum with needed protection against the copying of program code or the artistic elements of screen displays, while allowing the pace of innovation in the software industry to continue unimpeded.

CONCLUSION

The judgment of the Court of Appeals for the First Circuit should be affirmed.

Respectfully submitted,

DIANE MARIE O'MALLEY
Counsel of Record
MABEL NG
HANSON, BRIDGETT, MARCUS,
VLAHOS & RUDY
333 Market Street, Suite 2300
San Francisco, CA 94105-2173
(415) 777-3200
Attorneys for Amicus Curiae
SOFTWARE FORUM

December 1995

(19)

No. 94-2003

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

In the Supreme Court

OF THE

United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF AMICUS CURIAE OF PROFESSORS PETER S. MENELL AND DENNIS S. KARJALA IN SUPPORT OF RESPONDENT

PETER S. MENELL
(Counsel of Record)
Professor of Law
University of California at Berkeley
Boalt Hall School of Law
Room 472
Berkeley, CA 94720
(510) 642-5489

DENNIS S. KARJALA
Professor of Law
Arizona State University
College of Law
Box 877906
Tempe, Arizona 85287
(602) 965-4010

Attorneys for Amici
Peter S. Menell and Dennis S. Karjala

December 1995

BOWNE OF SAN FRANCISCO, INC. • 343 SANSOME ST., • S.F., CA 94104 • (415) 362-2300

BEST AVAILABLE COPY

37 pp

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
INTRODUCTION	1
ISSUE	2
SUMMARY OF ARGUMENT	2
ARGUMENT	4
COPYRIGHT IN A PROGRAM DOES NOT EXTEND TO THOSE "CERTAIN RESULTS" BROUGHT ABOUT BY PROGRAM EXECUTION	4
THE MENU COMMAND HIERARCHY DOES NOT CONSTITUTE COPYRIGHT SUBJECT MATTER	8
THE CRITICAL RELATIONSHIP BETWEEN COPYRIGHT AND PATENT LAW	11
APPLICATION OF THESE PRINCIPLES TO THIS CASE	21

TABLE OF AUTHORITIES

Cases

Page(s)

American Code Co. v. Bensinger,
282 F. 829 (2nd Cir. 1932) 20

Apple Computer, Inc. v. Microsoft Corp.,
799 F. Supp. 1006 (N.D. Cal. 1992), *aff'd*,
35 F.3d 1435 (9th Cir. 1994) 19, 29

Atari Games Corp. v. Nintendo of Am., Inc.,
975 F.2d 832 (Fed. Cir. 1992) 29

Atari, Inc. v. North Am. Philips Consumer Elecs. Corp.,
672 F.2d 607 (7th Cir.), *cert. denied*,
459 U.S. 880 (1982) 25

Baker v. Selden,
101 U.S. 99 (1879) ..... *passim*

Broderbund Software, Inc. v. Unison World, Inc.,
648 F. Supp. 1127 (N.D. Cal. 1986) 29

CMAX/Cleveland, Inc. v. UCR, Inc.,
804 F. Supp. 337 (M.D. Ga. 1992) 29

Cognotec Sers., Ltd. v. Morgan Guar. Trust Co.,
862 F. Supp. 45 (S.D.N.Y. 1994) 29

Computer Assocs. Int'l, Inc. v. Altai, Inc.,
982 F.2d 693 (2nd Cir. 1992) ..... *passim*

Continental Casualty Co. v. Beardsley,
253 F.2d 720 (2nd Cir.), *cert. denied*,
358 U.S. 816 (1958) 7

Digital Comms. Assocs. v. Softklone Distrib. Corp.,
659 F. Supp. 449 (N.D. Ga. 1987) 29

Engineering Dynamics, Inc. v. Structural Software, Inc.,
26 F.3d 1335 (5th Cir. 1994), *supplemented* 46
F.3d 408 (5th Cir. 1995) 8, 18, 29

TABLE OF AUTHORITIES (CONT'D)

CASES

Page(s)

Feist Pubs., Inc. v. Rural Tel. Serv. Co.,
499 U.S. 340 (1991) 6, 7, 17, 19, 20

Gates Rubber Co. v. Bando Chem. Indus., Ltd.,
9 F.3d 823 (10th Cir. 1993) 29

Graham v. John Deere Co.,
383 U.S. 1 (1966) 17

Harper & Row Pubs., Inc. v. Nation Enters.,
471 U.S. 539 (1985) 17

Hartfield v. Herzfeld,
60 F.2d 599 (S.D.N.Y. 1932) 20

Hartfield v. Peterson,
91 F.2d 998 (2nd Cir. 1937) 20

Hoehling v. Universal City Studios, Inc.,
618 F.2d 972 (2nd Cir.), *cert. denied*,
449 U.S. 841 (1980) 7

Jeweler's Circular Co. v. Keystone Pub. Co.,
281 F. 83 (2nd Cir.), *cert. denied*,
259 U.S. 581 (1922) 20

Kepner-Tregoe, Inc. v. Carabio,
203 U.S.P.Q. 124 (E.D. Mich. 1979) 16

Landsberg v. Scrabble Crossword Game Players, Inc.,
736 F.2d 485 (9th Cir.) *cert. denied*,
469 U.S. 1037 (1984) 7

Litchfield v. Spielberg,
736 F.2d 1352 (9th Cir. 1984), *cert. denied*,
470 U.S. 1052 (1985) 10

Lotus Dev. Corp. v. Borland Int'l, Inc.,
49 F.3d 807 (1st Cir. 1995) 9, 27

Lotus Dev. Corp. v. Borland Int'l, Inc.,
799 F. Supp. 203 (D. Mass. 1992) 7, 22, 23, 24, 25

TABLE OF AUTHORITIES (CONT'D)

CASES

Page(s)

Lotus Dev. Corp. v. Paperback Software Int'l., 740 F. Supp. 37 (D. Mass. 1990) 11, 29

Miller v. Universal Studios, Inc., 650 F.2d 1365 (5th Cir. 1981) 7

Mitek Holdings, Inc. v. Arce Eng'g Co., 864 F. Supp. 1568 (S.D. Fla. 1994) 8, 29

Mitel, Inc. v. Iqtel, Inc., 896 F. Supp. 1050 (D. Colo: 1995) 8, 9

Productivity Software Int'l, Inc. v. Healthcare Technologies, Inc., 1995 U.S. Dist. LEXIS 10381, Copy. L. Rep. (CCH) ¶ 27,440 (S.D.N.Y. 1995) 8, 29

Reiss v. National Quotation Bureau Inc., 276 F.717 (D.N.Y. 1921) 20, 21

Rosemont Enters., Inc. v. Random House, Inc., 366 F.2d 303 (2nd Cir. 1966), *cert. denied*, 385 U.S. 1009 (1967) 7

Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) 19, 29

Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2nd Cir.), *cert. denied*, 298 U.S. 669 (1936) 7

Stern Elecs., Inc. v. Kaufman, 669 F.2d 852 (2nd Cir. 1982) 25

Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978) 26

Whelan Assocs., Inc. v. Jaslow Dental Lab. Inc., 797 F.2d 1222 (3rd Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 6, 28, 29

TABLE OF AUTHORITIES (CONT'D)

Statutes

Page(s)

17 U.S.C. § 101 ..... 4, 10, 13

17 U.S.C. § 102 ..... *passim*

35 U.S.C. § 101 10

35 U.S.C. § 103 10

H.R. Rep. No. 94-1476, 94th Cong., 2d Sess. 53 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5666 9

Miscellaneous

Kenneth W. Dam, *Some Economic Considerations in the Intellectual Property Protection of Software*, 24 J. L. STUDIES 321 (1995) 10

Duncan Davidson, *Common Law, Uncommon Software*, 47 U. PITT. L. REV. 1037 (1986) 4

Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33 (1987) 15

DENNIS S. KARJALA, *Copyright Protection of Computer Software, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975 (1994) 5, 10

Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045 (1989) 15, 21, 28

Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329 (1987) 15, 28

Peter S. Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 COLUM. L. REV. 2644 (1994) 15

TABLE OF AUTHORITIES (CONT'D)

MISCELLANEOUS

	Page(s)
Arthur W. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977 (1993)	10
National Commission on New Technological Uses of Copyrighted Works, FINAL REPORT (1978)	4, 18, 19, 27
M. B. Nimmer & David Nimmer, NIMMER ON COPYRIGHT § 3.01 (1991)	10
Luca Paciola, SUMMA DE ARITHMETICA, GEOMETRIA, PROPORTIONI ET PROPORTIONALITA (1494)	21
Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of* Lotus v. Paperback, 6 HIGH TECH. L.J. 209 (1992)	6
U.S. Congress, Office of Technology Assessment, COMPUTER SOFTWARE AND INTELLECTUAL PROPERTY: BACKGROUND PAPER (March 1990)	27
U.S. Congress, Office of Technology Assessment, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY AND THE CHALLENGE OF TECHNOLOGICAL CHANGE (May 1992)	27

INTRODUCTION

The signatories to this brief are law professors at the University of California (Berkeley) and Arizona State University, respectively, who teach and write on copyright issues, and specifically on the proper application of copyright law to the protection of computer software.[1] We have received consent to file this *amicus* brief from both parties. We submit our views on what we perceive to be the crucial issue in this case because we believe that the District Court fundamentally misconstrued both the traditional copyright principles applicable to these facts and the role played by copyright in the overall system of intellectual property protection in the United States. As a result, the District Court effectively overruled the Supreme Court's decision in *Baker v. Selden*, 101 U.S. 99 (1879), a decision that for over 100 years has served to demarcate the boundary between the patent and copyright regimes.

The Court of Appeals for the First Circuit correctly applied section 102(b) of the Copyright Act to the facts of this case to deny copyright protection in the menu command hierarchy of the Lotus 1-2-3 computer program. While the First Circuit probably should have been more explicit in addressing whether it was treating the menu command hierarchy as a nonliteral element of the computer program or as a separate work, its ultimate conclusion is sound: However the menu command hierarchy might be classified as potential copyright subject matter, section 102(b) denies copyright protection. The result of the District Court's decision would be to extend copyright protection far beyond its traditional bounds—an expansion of protection that only Congress is authorized to make. It is therefore imperative

[1] We neither represent nor have represented any party to this action nor do we act on behalf of any person other than ourselves. We offer our views to the Court in the spirit of *pro bono publico*. Neither Borland nor its counsel has provided any financial remuneration in this matter whatsoever to amici other than to defray the costs of printing this brief.

that the Supreme Court affirm the First Circuit's reversal of the District Court's decision.

ISSUE

Properly viewed, this case does not involve any issue of the copyright protection of computer programs. Rather, it involves the application of traditional copyright law to the *products* of programs—those "certain results" produced by the combinations of statements or instructions that comprise computer programs. The issue in this case is whether functional aspects of these products, traditionally protected solely by patent or trade secret law under the doctrine of *Baker v. Selden*, 101 U.S. 99 (1879), should now be brought under the copyright umbrella. Stated otherwise, the issue is whether the particular spreadsheet format and methods of operation that constitute the user interface of the Lotus 1-2-3 computer program should be protected for the 75-year term of copyright solely on the ground that a variety of other spreadsheet formats and methods of operation are possible, or whether unpatented aspects of this technology should be permitted to progress unencumbered and incrementally through the cumulative efforts of many spreadsheet developers.

SUMMARY OF ARGUMENT

The argument that Borland has the right to create an independently written computer program that brings about the same results as the Lotus 1-2-3 computer program is two-fold: First, the statutory definition of a program distinguishes between the program instructions and what those instructions do—the "certain results" they bring about—when they are executed by a computer. The copyright in the program does not extend to the "certain results" brought about by the program. If those "certain results" are to be copyright protected, they must independently qualify as works of authorship. This analysis raises then a narrow but

potentially determinative subissue of whether the menu command hierarchy in question in this case constitutes copyright subject matter.

Second, even if aspects of the Lotus 1-2-3 user interface qualify as works of authorship independently of the underlying computer program—for example, as graphic or audio-visual works—the traditional limits on the scope of any copyright in that interface still apply. These limits include not only the copyright doctrines of merger and scènes á faire but also the foundational doctrines of *Baker v. Selden* and section 102(b) of the Copyright Act. These doctrines exclude function from the scope of copyright protection.

The policies underlying that tradition are and always have been firmly grounded in the fundamental allocation of protection undergirding the intellectual property system as a whole. Each of the intellectual property regimes—copyright, patent, trademark, and trade secret—must be understood in the context of this larger system and should not be interpreted in isolation. Moreover, only Congress, and not the courts, has authority to expand the scope of the intellectual property laws. Because we are addressing an asserted independent copyright in the outputs generated by the Lotus 1-2-3 program and not the program code itself, protection of functional features of the user interface are not governed by Congress's decision to protect some aspects of program code under copyright.[2] This case centers on the interpretation of *traditional* copyright principles applied to systems or methodologies for processing data and operating a computer through the medium of a computer program. Absent instructions from Congress to the contrary, courts must adhere to the requirements and limitations reflected in the Copyright Act, with due respect for the established case law interpreting the intellectual property statutes.

---

[2]See *infra* notes 13-14 and accompanying text.

ARGUMENT

COPYRIGHT IN A PROGRAM DOES NOT EXTEND TO THOSE "CERTAIN RESULTS" BROUGHT ABOUT BY PROGRAM EXECUTION

Under the Copyright Act, a computer program is "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (definition of "computer program"). Unless the "certain result" effected by a program is independently protected under copyright law, anyone has the right to produce independent code bringing about even an identical result. As emphasized by the National Commission on New Technological Uses of Copyrighted Works, FINAL REPORT (1978) (hereinafter referred to as "CONTU Report"), "One is always free to make the machine do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy." CONTU Report at 21. Even one of the strongest proponents of broad copyright protection for programs agrees that second comers are "free to emulate all external aspects of the program" without infringing the copyright. Duncan Davidson. *Common Law, Uncommon Software*, 47 U. PITT. L. REV. 1037, 1080 (1986). It is widely recognized, and not questioned in this case, that different computer programs can generate identical outputs. Therefore, even identity of unprotected output in no way establishes infringement of the underlying program.[3]

[3] In fact, to the extent that a specific piece of code is the only way of bringing about some aspect of this "certain result", one of only a few practical ways of doing so, or an obvious or standard way of doing so from the point of view of an experienced programmer, even that specific piece of *code* is not protected, and a finding of infringement cannot be based on similarities in these aspects of the code. *See* CONTU Report at 20 (first emphasis added):

> [C]opyright protection for programs does not threaten to block the use of ideas *or program language previously developed by others when that use is necessary to achieve a certain result.* When other language *is* available, programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works . . . .

This also follows, of course, from the filtering analysis called for by *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2nd Cir. 1992).

Software interfaces, including user interfaces, are an important part of that "certain result" that the set of instructions accomplishes. They are the doors and windows through which users as well as other hardware and software communicate with the program. The program designer must decide what kinds of interfaces are desired and then write code—the set of statements or instructions—to implement those decisions. Of course, the program also achieves other "certain results" in response to the user's inputs of commands and data, but the precise commands required as well as the types and formats for data that the program can "understand" are all features that are determined by, rather than an inherent part of, the set of statements or instructions that constitutes the copyright-protected computer program. Anyone wishing to change an interface, for example, could only do so by rewriting the code to bring about that result. *See generally* Dennis S. Karjala, *Copyright Protection of Computer Software, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975, 990-91 (1994).

This straightforward conclusion—that what a program *does*, and therefore its "appearance" to the outside world when viewed solely through input and output devices, is not protected by the copyright in the program—comports with well over 100 years of copyright tradition and was explicitly recognized by the Second Circuit in *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 703 (2nd Cir. 1992):

> [W]e note that our decision here does not control infringement actions regarding categorically distinct works, such as certain types of screen displays. These

items represent products of computer programs, rather than the programs themselves, and fall under the copyright rubric of audiovisual works.[4]

Therefore, the analysis of *Computer Associates* and other cases dealing with the scope of copyright protection for nonliteral elements of program code simply does not address the central issue of *Lotus v. Borland*: the protection of computer program outputs. The scope of such protection is circumscribed by all of copyright's traditional limiting doctrines and is not affected by the 1980 amendments to the Copyright Act recognizing the protection of computer programs—sets of statements or instructions—under copyright.[5]

---

[4]For a more detailed explication of the distinction between a computer program and its user interface, including some of the early judicial and administrative errors in addressing the issue, see Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of* Lotus v. Paperback, 6 HIGH TECH. L.J. 209, 264-69 (Appendix) (1992).

[5]Even if the Court were to approach the issue of copyright protection for interfaces as a part of the copyright in the underlying program, it is clear that what a computer program does—the screen displays it generates and the methodology it presents to the outside world for using the program via such means as keyboards and data entry formats—lies at a higher level of abstraction above literal code than even program structure and organization. The recognition of protection for so-called "structure, sequence, and organization" or "SSO" by the Third Circuit in *Whelan Assocs., Inc. v. Jaslow Dental Lab. Inc.*, 797 F.2d 1222, 1248 (3rd Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) caused great controversy, and *Computer Associates* explicitly reduced the breadth of *Whelan*. *See, e.g.*, 982 F.2d at 711-12 (noting that *Feist Pubs., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991) implicitly undercuts the *Whelan* rationale and citing with approval critics of *Whelan's* sweeping scope of protection). Courts following *Computer Associates* are still working out the level of detailed program structure that might be covered by a program copyright. Whatever the ultimate resolution of that problem, however, user interfaces cannot be considered protected nonliteral elements of programs because they exist at a level of abstraction even higher than overall program structure. User interfaces are therefore protected (if at all) as other types of works or under other branches of intellectual property law. To the extent that Lotus relies on the copyright in its 1-2-3 program for protection of that program's user interface, therefore, its claim must fail. More fundamentally, traditional copyright law requires that outputs of computer programs be evaluated solely on the basis of their own merits as expressive works of authorship and not on the basis of any copyright in the program that produces them. For this reason, as well, the District Court's attempt to distinguish *Baker v. Selden* as a precedent predating the congressional mandate to protect computer programs, 799 F. Supp. at 215, Appendix to Petition to Writ of Certiorari ("Pet.App.") 127a, is without logical support.

It should also be noted that simply treating programs as "literary works" does not lead to the conclusion that their interfaces are protected by the program copyright. Lotus would presumably like to see computer programs treated under copyright as if they were novels or plays, which receive a broader scope of protection than histories, biographies, rule books, fact works, legal forms, and scientific works. *Compare Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49 (2nd Cir.), *cert. denied*, 298 U.S. 669 (1936) (play) *with Feist Pubs., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991) (factual compilation); *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485 (9th Cir.) *cert. denied*, 469 U.S. 1037 (1984) (game strategy); *Miller v. Universal Studios, Inc.*, 650 F.2d 1365 (5th Cir. 1981) (history); *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 980 (2nd Cir.), *cert. denied*, 449 U.S. 841 (1980) (historical theory); *Rosemont Enters., Inc. v. Random House, Inc.*, 366 F.2d 303 (2nd Cir. 1966), *cert. denied*, 385 U.S. 1009 (1967) (biography); *Continental Casualty Co. v. Beardsley*, 253 F.2d 720 (2nd Cir.), *cert. denied*, 358 U.S. 816 (1958) (legal form). Abstract elements of novels and plays, such as detailed plots, may be protected, whereas rule books and similar works are essentially protected only against literal copying of text or close paraphrases. *E.g., Landsberg* at 488. Even as a purely technical matter of traditional copyright interpretation, computer programs are more analogous to rule books, recipes, legal forms, or scientific works than to novels and plays. Copyright in a computer program, by analogy, would be limited to literal program code and close paraphrases of code. It would hardly reach detailed SSO, let alone the program interfaces. One need not, therefore, rely on policy considerations concerning the copyright protection of functionality to challenge the attempted treatment of programs as novels or plays.

THE MENU COMMAND HIERARCHY DOES NOT CONSTITUTE COPYRIGHT SUBJECT MATTER

Notwithstanding the clear distinction made by the Second Circuit in *Computer Associates* between the program and its products, a number of courts have attempted to apply the abstraction-filtration-comparison analysis of that decision to various aspects of user interfaces.[6] The Court of

---

[6]The District Court in *Mitek Holdings, Inc. v. Arce Eng'g Co.*, 864 F. Supp. 1568, 1576-77 (S.D. Fla. 1994), stated that the copyright in a program for making architectural drawings extended to the "computer-user interface." However, the *Mitek* court properly applied traditional copyright law to that interface to conclude that the menu structure was a process outside the scope of copyright protection, *id.* at 1580, and further applied such traditional doctrines as merger and *de minimis* similarity to find for the defendant. Similarly, the District Court in *Productivity Software Int'l, Inc. v. Healthcare Technologies, Inc.*, 1995 U.S. Dist. LEXIS 10381, Copy. L. Rep. (CCH) ¶ 27,440 (S.D.N.Y. 1995), accepted the general principle that the program copyright could extend to the graphical user interface (screen displays), *id.* at *7, but concluded that the only similarities between the two interfaces were dictated by efficiency or functionality and were therefore not protectible under copyright law, *id.* at *10-*20. Like the *Mitek* court, the Fifth Circuit has opined that a program copyright extends to the "computer-user interface." *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1341 (5th Cir. 1994), *supplemented* 46 F.3d 408 (5th Cir. 1995). This decision, however, relies heavily on the District Court opinion in *Lotus v. Borland*, *see id.* at 1345-46, and makes the same analytical mistake: it concludes that creativity in methodological or systemic features of a program interface may of itself lead to copyright protection. *Id.*

The user interface at issue in *Mitel, Inc. v. Iqtel, Inc.*, 896 F. Supp. 1050 (D. Colo. 1995), is on all fours with that involved in *Lotus v. Borland*. The programs in question operated "call controllers" for business telephones and provided the user with a set of input codes that were used to set various parameters of the system to meet the needs of the user. The plaintiff claimed copyright protection in its set of input codes. The parties did not dispute that a variety of combinations or sequences could have been used to implement the various choices available to the user. *Id.* at 1053. However, once chosen, the user was

Appeals for the First Circuit in the present case carefully distinguished between nonliteral copying of program code and literal copying of the menu command structure. On this basis, the court correctly found that the *Computer Associates* analysis did not apply to Lotus's menu command hierarchy. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 814-15 (1st Cir. 1995), Pet.App. 14a-15a. Nonetheless, the court did not determine whether the menu command hierarchy falls within the intended scope of copyright subject matter.

The program, to be sure, is a literary work. The screen displays, moreover, can sometimes independently stand as graphic works, the help menus as literary works, and perhaps some of the data files as compilations. The menu command hierarchy, however, does not fit comfortably as an independent work within any of the classes of works set forth in section 102(a) of the Copyright Act (literary works, musical works, etc.). Although Congress intended the actual list of categories as "illustrative and not limitative," *H.R. Rep.* No. 94-1476, 94th Cong., 2d Sess. 53 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5666, Congress certainly intended section 102(a) as a whole to provide bounds on copyright subject matter. A computer program "menu command hierarchy" is not even roughly analogous to any of the section 102(a) categories of copyright subject matter. It is a method of operating a machine that is under the control

---

required to use the commands presented by the program. Without relying on the First Circuit's opinion in *Lotus v. Borland*, the *Mitel* court concluded that the command codes were simply a "procedure, process, system, and method of operation by which the customer can match the call controller functions to the long-distance carriers' technical needs and the end-user's choices." *Id.* at 1055. Even on the assumption that the codes were a part of the computer program—an argument that even the plaintiff did not press—the District Court also concluded that the command codes were an unprotectible "means to access or operate the program." *Id.*

of a computer program and is essentially a process that is at the heart of patent subject matter. 35 U.S.C. § 101. It is not a collection of preexisting materials or of data so as to constitute a "compilation" within the statutory definition. 17 U.S.C. § 101 (definition of "compilation"). Neither is it a nonliteral element of the screen displays, any more than Selden's system for using his accounting forms was a nonliteral element of those particular graphic works.

It is true that the menu command hierarchy represents a collection of functions available to the user and the methodology for implementing those functions. However, it cannot be the case that any work manifesting human intellectual creativity is copyright subject matter. Every patented invention, whether machine or process, necessarily manifests human intellectual creativity by reason of patent law's nonobviousness requirement. 35 U.S.C. § 103. An automobile, for example, may equally be considered a collection of individual functions and features, yet no one has seriously argued that an automobile is therefore a copyright-protected compilation.[7]

Consequently, a narrow ground for deciding this case is simply that the menu command hierarchy at issue is not copyright subject matter at all. However, because many user-interface disputes have involved, and will continue to involve, screen displays,[8] and because the limitations of

[7] Any argument that Borland infringed a section 106(2) right to create derivative works must also fail. A second work does not infringe the section 106(2) right unless it contains protected elements of the original work. *See* 1 Melville B. Nimmer & David Nimmer, NIMMER ON COPYRIGHT § 3.01(1991) ("If that which is borrowed consists merely of ideas and not of the expression of ideas, then although the work may have in part been derived from prior works, it is not a derivative work"); *Litchfield v. Spielberg*, 736 F.2d 1352, 1357 (9th Cir. 1984), *cert. denied*, 470 U.S. 1052 (1985).

[8] Infringement of the Lotus 1-2-3 screen displays was not an issue before the Court of Appeal. 49 F.3d at 812, Pet.App. 9a-10a.

Baker v. Selden apply to the functional aspects of screen displays with the same force as to those of the menu command hierarchy, the Court may wish to consider the application of those limitations in this case, assuming hypothetically that the menu command hierarchy does somehow constitute copyright subject matter.

THE CRITICAL RELATIONSHIP BETWEEN COPYRIGHT AND PATENT LAW

While this case should be decided under traditional copyright doctrine, an understanding of the critical relationship between copyright and patent protection is essential to understanding why a decision significantly limiting copyright protection for some user interfaces such as that produced by the Lotus 1-2-3 program best comports with the overall system of intellectual property protection. In fact, it is the broad copyright protection for such interfaces conferred by the District Court in this and its earlier *Paperback* decision, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990), Pet. App. 183a, that constitutes a radical departure from fundamental intellectual property law principles. *See* Dennis S. Karjala, *Copyright Protection of Computer Software, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975, 976-83 (1994).

Despite the similar basic approach of patent and copyright law—both draw a balance between providing an incentive for the creation of works and ensuring that the public and later creators can enjoy and build upon an expanding "public domain"—they are distinct components of a coherent scheme of intellectual property protection. Patent law seeks to promote the advancement of technology while copyright law seeks to encourage culture and the arts. Given their different foci, it is not surprising that Congress has crafted very different balances in these two statutes. In exchange for protection of claimed functional attributes of an invention, the patent law requires an inventor to demon-

strate to a technically trained examiner that the asserted invention is novel, nonobvious, and useful. After meeting these exacting requirements, the patent holder is entitled to 20 years of exclusive rights to practice the invention (measured from the date of filing). By contrast, the copyright law presents a much lower threshold for receiving protection—a minimal level of "creativity"—and foregoes any requirement of examination. In return, the author is entitled to protection limited to the expressive content of the work. The term of this protection, however, extends for the life of the author plus 50 years (or 75 years for entity authors).

In view of the very different thresholds for as well as scope and terms of protection, to extend copyright by protecting functional[9] works without explicit congressional authorization would undermine the integrity of the entire intellectual property law system. Thus, in the seminal case

---

[9]A number of commentators have argued that functionality is not a barrier to copyright protection by equating the term "functional" to "useful." *E.g.*, Kenneth W. Dam, *Some Economic Considerations in the Intellectual Property Protection of Software*, 24 J. L. STUDIES 321, 323-24 (1995) (pointing to copyright protection for dictionaries, maps, and charts and for the output of new technologies, such as photographs); Arthur W. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977, 986 (1993) (pointing to factual compilations, dictionaries, code books, encyclopedias, advertising, and instruction manuals). This equation of "usefulness" with "functionality," however, leaves no distinction between patent and copyright subject matter and forces the question of why we should have two very different statutory schemes for protecting the same thing.

In fact, it is clear that traditional copyright law, following the dictates of *Baker v. Selden*, has eschewed the protection of function, provided the meaning of the term "function" is understood in a precise way. Copyright does, of course, protect many works that are "useful" to human beings. Maps enable us to go from one place to another; recipes tell us how to bake cakes; accounting books explain how to implement a particular system of accounting. All of these works are copyright

of *Baker v. Selden*, 101 U.S. 99 (1879), decided more than a century ago, the Supreme Court explicitly recognized the

---

protected. They are not, however, "useful articles" within the definition in the Copyright Act:

> A "useful article" is an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. . . .

17 U.S.C. § 101. Although this definition was adopted in an effort to deal with the problem of industrial design—by excluding utilitarian functions other than to inform or portray an appearance from copyright protection and excluding even expressive aspects of useful articles to the extent they are not separable from the utilitarian functions, 17 U.S.C. § 101 (definition of "pictorial, graphic, and sculptural works")—it represents a statutory description of the kind of "usefulness" or "functionality" that has always been excluded from copyright protection, as opposed to the "usefulness" inherent in maps and recipe books that have long been a part of copyright. The code book cases, too, provide no basis for the copyright protection of functionality in this sense. *See infra* note 15.

The Lotus 1-2-3 user interface is functional in the sense used here because it has the intrinsic functional goal of permitting users of the program to input and manipulate data in a fast, efficient, and easy-to-master manner. If, by some chance, Lotus has managed to discover the fastest, most efficient, and easiest-to-master means of using spreadsheet programs, recognition of a copyright in its user interface is equivalent to giving Lotus a 75-year patent in an optimal functional tool. If, as is more likely, Lotus has made an important advance toward the goal of an optimal spreadsheet interface, recognition of a copyright in its functional features inhibits, for the same period, further incremental development in these tools, without any showing that the advance would have qualified for the shorter term of patent.

Of course, computer program *code* is also functional, and yet Congress has brought it under copyright protection. As we discuss below, however, the congressional decision to protect code under copyright does not imply that function ceases to limit the scope of protection for works other than code. *See infra* note 14.

fundamentally different roles played by copyright and patent law within that system:

> The copyright of the book, if not pirated from other works, would be valid without regard to the novelty, or want of novelty, of its subject-matter. The novelty of the art or thing described or explained has nothing to do with the validity of the copyright. To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters patent, not of copyright. The claim to an invention or discovery of an art or manufacture must be subjected to the examination of the Patent Office before an exclusive right therein can be obtained; and it can only be secured by a patent from the government.

101 U.S. at 102. The Supreme Court in *Baker* thus adopts a fundamental channeling principle that it then applies to the facts of the case:

> ... Charles Selden, by his books, explained and described a peculiar system of book-keeping, and illustrated his method by means of ruled lines and blank columns, with proper headings on a page, or on successive pages. Now, whilst no one has a right to print or publish his book, or any material part thereof, as a book intended to convey instruction in the art, any person may practice and use the art itself which he has described and illustrated therein. The use of the art is a totally different thing from a publication of the book explaining it. The copyright of a book on book-keeping cannot secure the exclusive right to make, sell and use account books prepared upon the plan set forth in such book. Whether the art might or might not have been patented, is a question which is not before us. It was not patented, and is open and free to the use of the public.

> And, of course, in using the art, the ruled lines and headings of accounts must necessarily be used as incident to it.

Id. at 104.

The fact that Selden's accounting forms may have satisfied the minimal level of "creativity" required by the copyright law and could have been expressed in a variety of ways was irrelevant to the *Baker* decision. A copyright in the book in no way prevents others from using the methods described, or the forms needed to execute the system.[10] In this way, the Supreme Court established a critical limitation on copyright protection to ensure that technological advances not satisfying the exacting requirements of patent law do not indirectly receive protection through copyright and to preserve the integrity of the entire intellectual property system.

The policy basis for the distinct approaches of patent and copyright law is the social desirability of allowing later technological creators—creators of functional works—to build on and improve, often in small ways, the earlier works of others. *See* Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33, 39 (1987); Peter S. Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 COLUM. L. REV. 2644, 2645-46 (1994) (discussing the relevance of sequential innovation to intellectual property protection); Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1057 (1989) (describing the importance of sequential innovation in the development of spreadsheet technology); Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329, 1338 (1987) (highlighting the role of sequential innovation

[10] Many cases have implemented this aspect of *Baker v. Selden*. For a more extended discussion and citations, see Pamela Samuelson, *supra* note 4, at 226-27 & n.73.

in computer technology). Technological improvements are often substantially similar to the products they improve and would infringe if the copyright standard were applied. Such improvements do not infringe a patent in the technology unless they adopt all of the elements, or their substantial equivalents, of a patent claim. Authors of copyright-protected works, as well, build upon public domain works and unprotected elements of protected works, but in taking from the latter, copyright law limits them to the more general or abstract features, together with those aspects denied copyright protection under *Baker v. Selden*, section 102(b), and other limiting doctrines.

Later inventors thus can apply a "ground up" approach to reliance on protected works that precede their own efforts. Once they are outside all patent claims, they are safe. Later authors, however, must adopt a "top down" (substantial similarity) approach and take a serious risk of being held liable for infringement as their reliance becomes more detailed. The different foci of patent and copyright law explain their different approaches to infringement. The social utility of allowing subsequent authors to make minor variations on a copyright-protected novel is minimal. For works of fiction, art, and music, variety is the spice of both legal and real life. We prefer to have one hundred different war novels than one hundred versions of *War and Peace* that differ only in their final chapter. Consequently, the broad scope of copyright protection for novels and the long period of its duration fulfills the goal of recognizing the author's creativity without unduly hindering later authors or depriving society of desirable works. *Kepner-Tregoe, Inc. v. Carabio,* 203 U.S.P.Q. 124, 131 (E.D. Mich. 1979) ("[T]here is no societal interest in many variants on a single theme or plot, nor is there the likelihood that by extending broad protection, entry to the market for literary works will be foreclosed").

Technology, by contrast, often develops incrementally, as later inventors add a bell or a whistle to an earlier invention to make it more desirable or useful to consumers. Many improvements on existing products, being rather straightforward or "obvious" in the sense of patent law, are given no intellectual property protection once they are released to the public. Such products often show at least as much intellectual creativity as many copyright-protected works, but their creators have a monopoly only for the period that is required for competitors to recognize the value or popularity of the improved product, figure out its "secret," if any, and gear up for production and marketing. In the case of technological products Congress has drawn the social policy balance at a different point than for traditional works of authorship because it is believed that to grant intellectual property rights in ordinary engineering advances would hinder the development of more and better products than it would encourage.[11] *Graham v. John Deere Co.*, 383 U.S. 1 (1966). Hence the "nonobviousness" requirement of patent law, as well as its shorter term and disclosure requirements, recog-

---

[11]The effort and creativity that go into unpatentable technological advances can be compared to similar unprotected efforts of historians and other authors:

> The urge to compensate for subsequent use of information and ideas is perhaps understandable. An inequity seems to lurk in the idea that much of the fruit of the historian's labor may be used without compensation. This, however, is not some unforeseen byproduct of a statutory scheme intended primarily to ensure a return for works of the imagination. Congress made the affirmative choice that the copyright laws should apply in this way.... To ensure the progress of arts and sciences and the integrity of First Amendment values, ideas and information must not be freighted with claims of proprietary right.

Harper & Row Pubs., Inc. v. Nation Enters., 471 U.S. 539, 589-90 (1985) (Brennan, J., dissenting), *quoted with approval in Feist Pubs., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991).

nize and promote the incremental nature of technological innovation.[12]

Computer programs, of course, at least in object-code form, are technological in nature: They sequentially set the switches inside a computing machine in such a way that the results can be interpreted by human beings as "information processing." To the extent that code is now copyright protected, Congress has indicated that a functional work should come under the copyright umbrella. The reason, however, is the vulnerability of program *code* to piracy. This was made clear by the National Commission on New Technological Uses of Copyright Works (CONTU)—established by Congress "to assist the President and Congress in developing a national policy for protecting both the rights of copyright owners and ensuring public access to copyrighted works when they are used in computer . . . systems, bearing in mind the public and consumer interest," CONTU Report at 3. The CONTU Report concluded that literal copying of computer programs would dramatically reduce the development costs of second comers and thereby create a disincentive to invest in computer program development.[13] Even if we assume, therefore, that the application of

[12]The Fifth Circuit in *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1346 (5th Cir. 1994), *supra* note 6, unwittingly recognized the importance of incremental improvement in the context of user interfaces by finding creativity in plaintiff's interface partially on the basis that defendant's program managed to perform the same functions with significantly fewer input formats. It should have realized that creativity in plaintiff's unpatented but functional product could not be a basis for a long-term prohibition under copyright of defendant's further exercise of creativity in improving the product.

[13]CONTU Report at 11. Every concrete example of infringement that CONTU offers involves direct and literal copying, such as photocopying of printed source code and one-to-one transcription from magnetic tape or disk to paper. CONTU Report at 22-23. The Report goes on to state that "[m]ost infringements, at least in the immediate future, are likely to involve simply copying," *Id.* at 22, but the problem it predicts for the

Baker v. Selden and section 102(b) to program code is now more limited than to other technological works—in that the literal code of every program can be said to implement a process or method of operation and yet remains copyright protected—there is no basis for concluding that Congress intended to remove any barriers to the copyright protection of function in other classes of works.[14] The Supreme Court in *Feist*, 499 U.S. at 350 & 364, the Second and Ninth Circuits in *Computer Associates*, 982 F.2d at 703-05, and *Sega Enterprises Ltd. v. Accolade*, Inc., 977 F.2d 1510, 1524 (9th Cir. 1992), *amended by* Order and Amended Opinion, D.C. No. CV-91-3871-BAC (Jan. 6, 1993), and the Northern District of California in *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1021 (N.D. Cal. 1992), *aff'd*, 35 F.3d 1435 (9th Cir. 1994), have recently reaffirmed the fundamental role of *Baker v. Selden* in copyright analysis.[15]

---

then-future was a technology that permits programs to be stated orally or permits use of a program without copying. The CONTU Report nowhere refers to copying of "nonliteral elements," much less copying of user interfaces or other results of programs, as potential infringement.

[14]The CONTU Report is clear that "[o]ne is always free to make the [computer] do the same thing as it would if it had the copyrighted work placed in it," CONTU Report at 21, and that the fundamental limitations of copyright reflected in *Baker v. Selden* and section 102(b) would remain in force. *Id.* at 18-20. There is no indication in the textual changes to the 1976 Act or the legislative history that Congress intended to extend copyright protection to any functional subject matter other than computer program code. *See supra* note 13 and accompanying text. To the contrary, the careful definition of "computer program" in section 101, distinguishing between "a set of statements or instructions" (*i.e.*, code) to be used in a computer and the "certain result[s]" (*e.g.*, user interface) such instructions bring about, reinforces this interpretation of the Copyright Act. In view of the obvious tension with the patent law of a more expansive interpretation, the Court should not engraft broad new protections for functional works upon the Copyright Act.

[15]Lotus cites the code book cases in support of its claim for copyright protection of its user interface. Brief for the Petitioner at 23 n.32 & 25

n.37, *citing Reiss v. National Quotation Bureau, Inc.*, 276 F. 717 (D.N.Y. 1921); *Hartfield v. Peterson*, 91 F.2d 998 (2nd Cir. 1937); *American Code Co. v. Bensinger*, 282 F. 829 (2nd Cir. 1932); *Hartfield v. Herzfeld*, 60 F.2d 599 (S.D.N.Y. 1932). The *Bensinger* case would raise serious questions of originality under today's standards. All the plaintiff added to what he copied from his English source consisted of a large set of five-letter combinations that appear from the sample shown in the opinion to have been created simply by incrementing letters of the alphabet in the same way that one increments numbers. 282 F. at 833 n.1. Copyright protection was based solely on *Jeweler's Circular Co. v. Keystone Pub. Co.*, 281 F. 83 (2nd Cir.), *cert. denied*, 259 U.S. 581 (1922). *Id.* at 833. *Jeweler's Circular* and the "sweat of the brow" theory of originality for which it became the basis was thoroughly repudiated by the Supreme Court in *Feist*. 499 U.S. at 352-53, 359-60. The question of copyright protectibility of codes was not argued in the *Herzfeld* case, which in any event was heavily influenced by the "sweat of the brow" reasoning of the Second Circuit precedents by which the District Court was bound. 60 F.2d at 600 (["I]t must be recognized that the plaintiff's basic work . . . was the result of years of labor"). The *Peterson* court found copyright protection for the collection of words as a compilation and came to the astounding conclusion that copying of *anything* from a protected compilation was infringement. 91 F.2d at 1000 ("When the statute allows a compilation to be copyrighted, it seems clear that no one can copy phrases or sequences which are original with the author or appropriate any other part of the copyrighted work, whether that part is in the public domain or not").

In any event, the code book cases are not authority for the copyright protection of functionality, provided the term "function" is understood properly. *See* note 9 *supra*. Like *Reiss*, these cases hold at most that a book of meaningless code words may be copyright protected. That does not make the works functional in the sense that distinguishes patent from copyright. Under today's statutory definition of a "useful article," the issue is whether such a list of code words has an intrinsic utilitarian function other than to portray themselves or to convey information. Such meaningless words, by definition, do not in themselves convey information, and their only apparent function is simply to portray themselves when meanings and/or syntax is added to make a secret code. Had the code books used newly created hieroglyphic-type characters, few would doubt the copyright protectibility of the characters as pictorial works (or deny their status as a useful article), because the only function of the characters would be to portray themselves (whether or not the user adds meanings or syntax). The code words in *Reiss* and the other code book cases portray themselves as letter groups rather than pictures, but they have no other utilitarian function, intrinsic or otherwise. Consequently, they are not "functional" in the sense that distinguishes patent and copyright subject matter. This was obvious to Judge Hand in *Reiss*: "*Baker v. Selden* . . . is too foreign to the case at bar to deserve comment." 276 F. at 719.

APPLICATION OF THESE PRINCIPLES TO THIS CASE

On a technical analytical level, a court could hardly face a situation in the modern age more squarely on point with *Baker v. Selden* than *Lotus v. Borland*. Selden had created an accounting system—building on ideas of others dating back at least to the innovation of double-entry bookkeeping in 1494 by Luca Paciola[16]—that employed "ruled lines and blank columns, with proper headings on a page, or on successive pages." 101 U.S. at 104. The organization and content of the blank forms were necessary for anyone to use Selden's system. The forms were, in other words, the predigital interface between Selden's system and the user. Just as Selden's accounting system was an incremental improvement upon prior systems of accounting, Lotus freely built upon prior electronic spreadsheet systems dating back at least to Visicalc, introduced in 1979.[17] Like Selden, Lotus

[16]See Luca Paciola, SUMMA DE ARITHMETICA, GEOMETRIA, PROPORTIONI ET PROPORTIONALITA (1494), *discussed in* T. BUDD & E. WRIGHT, THE INTERPRETATION OF ACCOUNTS 7 (1930).

[17]*See* Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STANFORD L. REV. 1045, 1057 (1989) (discussing the process of technological advancement in application programming). The basic methods underlying Visicalc and Lotus 1-2-3 extend back at least to the time of Luca Paciola, as a principal application of spreadsheet analysis is bookkeeping. The key distinctive feature of the Visicalc and Lotus products is the ability simultaneously to calculate and alter tables of information. The basic means by which this ability was brought into accounting—digital processing—was of course the product of neither Visicalc nor Lotus but

started with existing technology and created a system for entering information into a computer, processing the information, displaying the information on the screen, and putting results obtained into hard copy.[18] That system is implemented by a computer program, and because of the speed of digital processing it is more sophisticated than Selden's accounting system, but there is no essential distinction between them as far as traditional copyright law is concerned. In fact, the presentation of a spreadsheet is often remarkably similar to and no more complex than Selden's "ruled lines and blank columns, with proper headings on a page, or on successive pages." 101 U.S. at 104. Moreover, the menu structure of the Lotus system is a sequence of operations analogous to the ordered process of bookkeeping described in Selden's book.

In stark contrast to the Supreme Court's analysis in *Baker v. Selden*, the District Court in *Lotus v. Borland* begins its infringement analysis with an explicit refusal to treat as unprotected idea the very systemic aspects of the Lotus 1-2-3 interface that are denied protection under *Baker*. The Court sets up an abstractions scale going from an electronic spreadsheet at the high end down to the precise Lotus 1-2-3 command set as hierarchically arranged by Lotus at the low end. 799 F. Supp. at 216, Pet.App. 128a-

---

rather hundreds of predecessors, human and corporate, active in the development of computer science.

[18] What the *user* creates using the Lotus system may, of course, be copyright protected. The user-created products of most spreadsheet and word processing software almost always have no function other than to inform or entertain. They are therefore not "functional" in the sense used to distinguish patent and copyright subject matter. *See* note 9 *supra*. This product of the end user must be distinguished from the functional interface developed by Lotus to assist the user's creation of such works. The typewriter is a device that also assists in the creation of copyright-protected works, but the machine itself is not protected by copyright, nor is its user interface—the arrangement of the keys.

130a. It rejects Borland's argument that the latter is also unprotected for the following reasons:

> The premise of Borland's argument is that an "idea" of Lotus 1-2-3 version 2.01 is complete compatibility with earlier versions of 1-2-3, and more precisely with macros generated for use with earlier versions. Borland argues that the precise menu commands and menu structure are necessary to such functional compatibility. . . . This argument is essentially tautological. As applied to any case involving a useful article, an argument of this kind would always define the idea to incorporate all the specifics of the particular expression of that idea in the allegedly copyrightable work. Nothing would be copyrightable under this methodology of analysis.

799 F. Supp. at 216-17, Pet.App. 130a. Judge Keeton's analysis here is perfectly correct, as far as it goes. Nothing in the user interface of Selden's accounting system *was* copyright protected, precisely because of the necessity of using that specific interface in practicing Selden's system. Unfortunately, Judge Keeton draws exactly the wrong conclusion from his analysis. His insistence that *something* in the Lotus 1-2-3 system must be copyright protected leads to a result in direct conflict with the holding of *Baker v. Selden*.

The second step of the District Court's infringement standard compounds this error by simply assessing whether there were many ways of representing spreadsheet interfaces and designing menu trees for inputting and processing information:

> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or

> method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

799 F. Supp. at 217, Pet.App. 131a. The Court finds that by changing the names of the commands (and therefore the "natural" keystrokes used to invoke them, such as their initial letters) in the main menu and submenus, an "extremely large" number of possibilities different from the specific choices made by Lotus were possible. *Id.* at 217-18, Pet.App. 131a-133a. The Court adopts a similar approach with respect to the arrangement or hierarchy of the commands. *Id.* at 218-19, Pet.App. 132a-136a. Had it been dealing with a fanciful work, the District Court's analysis would have been appropriate, but *Baker v. Selden* and section 102(b) of the Copyright Act mean more than the merger doctrine for functional features. The whole point of *Baker's* channeling principle between patent and copyright law is that systems and methodologies are not copyright protected, even if creative and even if other systems or methodologies are available. In this case, users of Lotus 1-2-3 have learned the Lotus system and many want to use *it*, not some other system. Other users have developed "macros" that must combine Lotus commands precisely as Lotus defines them in order to run on the Lotus system. To use the Lotus system, there *must* be identity of commands and command structures. Of course, it is possible for Borland and other spreadsheet developers to create other systems. Borland, in fact, has done that with its native mode interface. But that is irrelevant to the application of *Baker v. Selden* to the facts of this case: Under *Baker*, the Lotus system is simply unprotected by copyright.[19]

---

[19]That the District Court did not apply the proper test in this case is clear. The Court repeatedly refers to functionality in elements it found to be protected: "[T]he menu commands are an important part of the functionality of the macros." 799 F. Supp. at 208, Pet.App. 112a. "Borland admits to copying the functionality of the keystroke sequences and macro language." *Id.* at 209, Pet.App. 115a. "The menu command hierarchy is an integral part of the functionality of the macros and of the keystroke sequences." *Id.* at 219, Pet.App. 134a.

Applying the District Court's analysis to the facts of *Baker v. Selden* would therefore require a different outcome in that case. There are countless ways of bookkeeping and methods for inputting and processing accounting information. Hence, Selden would be entitled to protection for his system as well as his forms for recording the information in the manner demanded by his system. Had Selden implemented his accounting system by way of a computer program, as Lotus did, rather than simply describe it in a book, he would under this analysis be entitled to protect the "ruled lines and headings of accounts" and structured methodology of his system. Thus, the District Court in this case effectively overruled *Baker v. Selden*, a central pillar in our intellectual property protection system.

This is not to say that there can never be copyright-protected "expression" in a computer program user interface. A straightforward example would be the fanciful characters and the environments in which they chase or avoid enemies in video games. *E.g., Stern Elecs., Inc. v. Kaufman*, 669 F.2d 852, 854 (2nd Cir. 1982) (spaceships, ground missile bases and fuel depots, and background terrain); *Atari, Inc. v. North Am. Philips Consumer Elecs. Corp.*, 672 F.2d 607, 617 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982) (Pac Man characters). It is even possible that some aspects of the Lotus 1-2-3 user interface, at least its screen displays, are included solely for aesthetic purposes and are not part of any of the processes, operations, or methods by which the user actually employs the program. Textual information under the "Help" operation will in many cases be copyright protected like any other literary work. But that should be the test: Is an aspect of the interface chosen solely to portray an appearance or to

convey information or does it have another utilitarian function—is it something that the user must do in order to make *this* program operate in the intended manner?[20] If it is, section 102(b) and *Baker v. Selden* require that it be excluded from copyright protection.

Lotus may feel that it has made a fundamental advance in making personal computers useful to nonspecialists. If, in fact, Lotus has made a nonobvious advance in the art over the products of predecessors on whom Lotus relied, it should have sought patent protection for those advances.[21] Lotus has a copyright in its program, which is undoubtedly a complex piece of work, that affords Lotus important protection against piracy of program code. Lotus does not allege, however, that Borland copied its program code or protectible nonliteral elements (if any) of that program code. Although it is not strictly relevant to the copyright analysis, Borland did not even gain the advantage over Lotus of avoiding the design costs of a graphical user interface for a spreadsheet program because Borland designed *both* what it felt would be a better interface *and* wrote code that would implement the Lotus interface to accommodate users who, like the beleaguered users of the QWERTY typewriter keyboard, had learned the Lotus system and were reluctant to change. It is precisely that kind of productive use of unpatented functional works that the exclusion from copyright protection of *Baker v. Selden* is designed to promote.

We also do not mean to suggest that aspects of user interfaces or other outputs of computer programs might not merit greater protection than currently exists under intellectual property law. Rather, we believe, as did CONTU, *see*

[20] *See* note 9 *supra*.

[21] It is worth noting that the Lotus 1-2-3 interface was itself developed at a time when the only judicial decision on the question pointed in the direction of nonprotection under copyright. *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003 (N.D. Tex. 1978).

CONTU Report at 46, that Congress is the appropriate body to expand the scope of intellectual property protection. In the 1980 amendments to the Copyright Act and the Semiconductor Chip Protection Act of 1984, Congress has, after careful study and broad involvement of affected interests and the public at large, extended intellectual property protection to new classes of computer-related works. In recent years, the Office of Technology Assessment has undertaken at the behest of Congress studies of the efficacy of existing legal protection for computer software. *See* U.S. Congress, Office of Technology Assessment, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY AND THE CHALLENGE OF TECHNOLOGICAL CHANGE (May 1992); U.S. Congress, Office of Technology Assessment, COMPUTER SOFTWARE AND INTELLECTUAL PROPERTY: BACKGROUND PAPER (March 1990). We fully expect that Congress will seek to address perceived gaps in this system should existing protection, properly applied by the courts, prove inadequate. It is not, however, the province of courts to take over this lawmaking function by extending protection beyond the limits established by traditional doctrines undergirding the intellectual property system. The District Court in *Lotus v. Borland* clearly exceeded these bounds, expanding protection afforded *outputs* of computer programs—user interfaces—and effectively overruling *Baker v. Selden* without the authorization of Congress. While perhaps not fully appreciating the implications of *Baker v. Selden* to the facts before it,[22] the First Circuit in this case equally clearly recognized the necessary application of sec-

[22] The First Circuit in this case concluded that *Baker* did not control because the screen displays of the Lotus 1-2-3 program were not at issue and because only those screen displays were analogous to the rows and columns of an accounting ledger. 49 F.3d at 814, Pet.App. 12a-13a. The menu command hierarchy that the First Circuit found to be a "method of operation" under section 102(b), however, is no less a system than Selden's sequence of accounting operations. In focusing too closely on

tion 102(b) of the Copyright Act—itself a partial codification of *Baker*—properly to reverse the District Court's decision. Even if additional protection were warranted for computer-user interfaces, it is not at all clear that traditional copyright protection—with its long term of protection and exclusivity of rights—would be appropriate. *See* Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STANFORD L. REV. 1329, 1370-71 (1987); Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STANFORD L. REV. 1045, 1050 n.20 (1989). As other courts grappling with software cases have recognized, the judiciary is ill-suited to weighing the complex issues involved in expanding or reforming intellectual property law. *See, e.g., Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2nd Cir. 1992).

Computer software, by its very nature as written work intended to serve utilitarian purposes, defies easy categorization within our intellectual property system. Cases involving computer software have therefore presented difficult challenges for courts, which have not always correctly applied traditional copyright doctrines in conjunction with Congress's adoption of CONTU's recommendations to a complex and evolving technology. As a prime example, *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3rd Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987), stood for six years as an influential decision on the scope of protection for nonliteral aspects of program code despite cogent academic criticism. The Second Circuit's sweeping repudiation of the *Whelan* approach in the *Computer Associates* case in 1992, building upon this academic literature and reasserting essential limiting doctrines of copyright law, has supplanted the

---

the precise factual holding of *Baker*, the First Circuit briefly lost sight of the vastly more important general principle for which *Baker* stands.

Whelan approach in substantially all subsequent decisions.[23] Similarly with regard to copyright protection for interfaces, most early district courts addressing the issue—resting upon an erroneous assumption that the copyright in the computer program extends to the outputs of the program and/or building upon the faulty foundation of *Whelan*—have gone astray of fundamental copyright doctrines.[24] At least one Circuit Court of Appeal has also lost its bearings in applying copyright to this new type of functional work.[25] This Court should affirm the First Circuit's decision, thereby ensuring that copyright protection for the outputs of computer pro-

---

[23] *Computer Assocs. Int'l Inc. v. Altai, Inc.*, 982 F.2d 693 (2nd Cir. 1992); *see e.g., Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 840-41 (10th Cir. 1993); *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524-25 (9th Cir. 1992), *amended by* Order and Amended Opinion, D.C. No. CV-91-3871-BAC, Jan. 6, 1993; *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *Productivity Software Int'l v. Healthcare Tech., Inc.*, 1995 U.S. Dist. Lexis 10381, *7, Copy L. Rep. (CCH) ¶ 27,440 (S.D.N.Y. 1995); *Cognotec Sers., Ltd. v. Morgan Guar. Trust Co.*, 862 F. Supp. 45, 49 (S.D.N.Y. 1994); *Mitek Holdings, Inc. v. Arce Eng'g Co.*, 864 F. Supp. 1568, 1577 (S.D. Fla. 1994); *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1024-25 (N.D. Cal. 1992); *CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337, 352 (M.D. Ga. 1992).

[24] *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990), Pet.App. 183a; *Digital Comms. Assocs. v. Softklone Dist. Corp.*, 659 F. Supp. 449 (N.D. Ga. 1987); *Broderbund Software, Inc. v. Unison World, Inc.*, 648 F. Supp. 1127 (N.D. Cal. 1986).

[25] *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335 (5th Cir. 1994), *supplemented* 46 F.3d 408 (5th Cir. 1995).

grams remains in line with the fundamental principles of the Copyright Act.

Respectfully submitted,

PETER S. MENELL
(Counsel of Record)
Professor of Law
University of California at Berkeley
Boalt Hall School of Law
Room 472
Berkeley, CA 94720
(510) 642-5489

DENNIS S. KARJALA
Professor of Law
Arizona State University
College of Law
Box 877906
Tempe, Arizona 85287
(602) 965-4010

Attorneys for Amici
Peter S. Menell and Dennis S. Karjala

December 1995

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

No. 94-2003

In The

Supreme Court of the United States

October Term, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL, INC.,

Respondent.

On Writ Of Certiorari
To The United States Court Of Appeals
For The First Circuit

BRIEF AMICUS CURIAE OF SOFTWARE INDUSTRY COALITION JOINED BY THE COMPUTER SOFTWARE INDUSTRY ASSOCIATION IN SUPPORT OF RESPONDENT

THOMAS F. VILLENEUVE
(Counsel of Record)
GUNDERSON DETTMER STOUGH
VILLENEUVE FRANKLIN &
HACHIGIAN, LLP
600 Hansen Way, Second Floor
Palo Alto, California 94304
(415) 843-0500

Attorneys for Amicus
Software Industry Coalition Joined by Others

December 1995

COCKLE LAW BRIEF PRINTING CO., (800) 225-6964
OR CALL COLLECT (402) 342-2831

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES ii

I. INTRODUCTION AND INTEREST OF AMICUS 1

II. SUMMARY OF THE ARGUMENT 2

III. ARGUMENT 4

A. The Decision of the First Circuit Maintains the Level of Copyright Protection Necessary to Encourage Software Development 5

B. The First Circuit's Decision Does Not Leave the Software Industry With Inadequate Intellectual Property Rights 9

IV. CONCLUSION 14

TABLE OF AUTHORITIES

Page

CASES

Ashton-Tate v. Ross, 916 F.2d 516 (9th Cir. 1990) 10n

Atari Games Corp. v. Nintendo of America, Inc., 975 F.2d 832 (Fed. Cir. 1992).....10n

Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 7n, 10n

Lotus Development Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) 8, 13

Sega Enterprises, Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1993).....10n

Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978) 10

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) 7n, 10

STATUTES

17 U.S.C. § 101 .....6n

17 U.S.C. § 102(b) 8

MISCELLANEOUS

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1978) 5, 6, 7, 13

Nelson R. Capes, *The Software Copyright "Super Patent,"* 11 Computer Lawyer at 8 (June 1995) ...12n

BRIEF AMICUS CURIAE OF SOFTWARE INDUSTRY COALITION JOINED BY THE COMPUTER SOFTWARE INDUSTRY ASSOCIATION

I. INTRODUCTION AND INTEREST OF AMICUS

The signatories below have received consent to file this brief from both Petitioner and Respondent.

The Software Industry Coalition ("Coalition") is a non-profit mutual benefit corporation that is dedicated to improving the quality and effectiveness of products, processes, and relationships in the software industry. Our mission is to identify specific projects and to build collaborative relationships to facilitate an industry consensus on software issues, to act as a catalyst in resolving those issues, and to present a coordinated software industry voice. Members of the Coalition include companies whose primary products are software, companies whose primary products are hardware for which software is a significant component, companies that provide information services, companies that provide significant professional services to software companies, and government technology laboratories.[1]

The Computer Software Industry Coalition ("CSIA") is a non-profit organization that monitors proposed legislation and regulation affecting the computer software industry, disseminates that information to the industry,

---

[1] Although respondent Borland International, Inc. is a member of the Coalition, it has abstained from the decision to file this brief. Borland is not a member of the Computer Software Industry Association. This brief was approved pursuant to the balloting procedures of the signatory organizations who exercised complete control over its editorial contents. Borland helped to defray the costs of printing this brief.

works to form industry consensus, and communicates industry positions to legislative bodies. CSIA represents over 3,000 companies and professional individuals in the software industry.

Although we in the software industry rely upon the copyright law as an essential means of protection for our investment in the development of new and beneficial software products, we are strongly of the view that the overextension of copyright protection would have severe adverse consequences for our industry. The decision of the First Circuit gave the industry the two things it desires most from the copyright law: an appropriate level of protection, and clear-cut guidelines to determine its boundaries. The Coalition files this brief to make known the perspective of the software industry on these issues, and to urge the Court to affirm the decision of the First Circuit.

II. SUMMARY OF THE ARGUMENT

The Coalition was extremely concerned about the implications of the District Court's decision. The District Court appeared to believe that it could not give software developers too much of a good thing, as if because Lotus had developed a good software product, copyright should be extended to protect even the functional features of that product. Such thinking is misguided.

Indeed, in our experience, progress in the software industry is jeopardized more by drawing the bounds of intellectual property protection too broadly than by drawing them narrowly. In the 1976 Copyright Act, and

in the 1980 amendments, Congress heeded the software industry's request for protection from copying of the code of computer programs. However, the industry did not seek – nor did Congress provide – protection for mere functional aspects of computer programs, including methods of operation and the like, such as the arrangement of menu commands. Such protection remains unwanted to this day, as amply reflected by the fact that of the thousands of true software developers in this country, not one has filed a memorandum in support of Lotus' position. Under such circumstances, Lotus' protestation of concern for protecting the "lifeblood" of the software industry (Brief for the Petitioner, p. 49) rings hollow.

The Coalition, by contrast, which has many developer members, is representative of the software industry. The Coalition does not believe, contrary to Lotus' claims, that the reasoning of the Court of Appeals in any way jeopardizes our continued ability to obtain needed copyright protection for our computer code.

To say that denial of protection for mere methods of operation will render software products unprotected by copyright is to misunderstand the true purpose of copyright protection of software. The creation of new and original expression to produce a given result in a software product is specifically permitted by the copyright laws. However, the greatest protection copyright affords for software is as a weapon against identical reproduction of the software code.

Software code is essentially comprised of pure digital information. Its identical reproduction is easily, inexpensively, and rapidly accomplished. The software industry

loses millions, even billions, of dollars yearly from such exact duplication and subsequent free or piratical distribution. "Pirates" are not bothering to create new and original expressions; they are merely copying the originals. Copyright is the essential means of proving and deterring such activity.

Thus, we believe that the decision of the First Circuit strikes the proper balance between giving sufficient copyright protection to protect against software piracy, while not stifling the development of innovation in the industry. Moreover, the decision sets forth a clear, discernible rule against which we in the industry can measure our conduct, to make reasonable predictions regarding which elements are, and which are not, subject to copyright protection. For all of these reasons, we urge affirmance of the decision below.

III. ARGUMENT

Both Lotus and its amici have stressed the dynamism and the importance of the American software industry. Lotus, for example, describes it as a "dynamic" and "vibrant" national industry. Brief for the Petitioner, p. 16, 49. The brief for DEC, Intel, Xerox, and Gates Rubber similarly describes computer software as "one of the fastest-growing sectors in our national economy," with the

United States holding a dominant 75% share of the $70 billion worldwide market.[2]

As members of the software industry, we agree that it is a vital component of the American economy. We believe, however, that its health would be impaired, not improved, by the copyright regime Lotus proposes.

A. The Decision of the First Circuit Maintains the Level of Copyright Protection Necessary to Encourage Software Development.

As computer software developers, we recognize the important role served by copyright in protecting our investment in the creation of computer programs. In particular, because the text of a computer program – the "code" – is difficult and costly to write, but easy and cheap to copy, we believe that copyright is critical in protecting against the unauthorized copying, distribution, and deployment of programs by software "pirates." Inadequate protection would serve as a major disincentive to software development by eroding the prospect of appropriate compensation to software developers through software licensing or sales.

These concerns were the principal focus of the hearings before the National Commission on New Technological Uses of Copyrighted Works ("CONTU"). *See* CONTU

---

[2] Brief of Digital Equipment Corporation, the Gates Rubber Company, Intel Corporation and Xerox Corporation in Support of Petitioner ("Hardware Cos. Brief"), p. 2.

Final Report at 11 (because "the cost of developing computer programs is far greater than the cost of their duplication," protection is required "to encourage the creation and broad distribution of computer programs"). Based on this concern, CONTU recommended, and Congress enacted, the 1980 amendment to the 1976 Copyright Act, which confirmed that computer program code is protectable under copyright.

Congress did not, however, either in 1976 or in 1980, extend this copyright protection for computer programs to mere functional aspects such as methods of operation, as the First Circuit recognized the Lotus menu command hierarchy to be. Not only does that structure fail to meet the statutory definition of "computer program,"[3] but the CONTU Final Report made it clear that copyright protection for the program must be determined separately from copyright for its output. (Output includes the on screen environment which is manifest by reason of the computer software's instructions, much as the printout from the computer's printer is manifest, except that the on screen "output" is usually manipulable, as is the case with the menu command hierarchy.)

In discussing the relationship between the program and its output, the Vice-Chairman of CONTU, Professor Nimmer, proposed that a program be copyrighted only when the program's output also qualified separately for

[3] Under the Copyright Act, a computer program is "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. In this definition, the program's output is the "certain result" that is effected by running the program.

copyright protection. However, the CONTU Final Report ultimately rejected that position, concluding that there should "be no distinction made between programs which are used in the production of further copyrighted works and those which are not." *Id.* at 21. Under this analysis, therefore, the program's output may – or may not – be a work that itself qualifies for copyright protection. The question of the output's qualification for copyright protection, however, in no way jeopardizes the protection that the developer has in the program itself.

In our view, the First Circuit's decision fully accords with this separate copyright treatment of the program and its output, respectively. Thus, the Court of Appeals' analysis of whether there is copyright protection in the arrangement of Lotus' menu tree is entirely independent of the viability of copyright protection for Lotus' 1-2-3 code.[4] Rather, the Court of Appeals correctly determined that the program's output, including the arrangement

[4] The Coalition expresses no view whether some level of abstraction away from the program's code – for example, what is often referred to as the "structure, sequence, and organization" of the code – is protected under copyright. *Compare Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1248 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) (recognizing such protection) *with Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 711-12 (2d Cir. 1992) (criticizing breadth of protection under *Whelan*). The issue simply is not presented in this case. No matter how high a level of abstraction one considers, the *Whelan/Altai* debate turns on the similarity at that level of the programs' *code*. Here, Lotus' claim is that Borland could not copy its menu screen output, even though there was nothing in common between the two programs at any level of abstraction. Indeed, Lotus did not even put its code into evidence. Trial Tr. of Apr. 1, 1993, p. 5-136 - 5-137, JA 300.

and structure of its menu tree, must be evaluated separately on its own merits. *Lotus Development Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 815-16 (1st Cir. 1995), Pet. App. 16a-18a. Because Lotus did not contend that Borland had copied the look of its screen displays, *id.*, the Court expressly did not reach the question whether the 1-2-3 screen displays, *per se*, enjoy copyright protection.[5] *Id.* at 815-16 & n.10, Pet. App. 16a. Similarly, the Lotus "long prompts" – the longer, more descriptive explanations of 1-2-3 commands – were not before the Court on appeal. *Id.* at 815 & n.9, Pet. App. 16a.

What remained was the question whether the words, order, and arrangement of the 1-2-3 menu tree constituted a copyrightable work. On this question, the Court of Appeals' analysis was clear and intuitive: the menu tree serves the same intrinsic functional purposes as the buttons of a VCR, and hence, like the buttons, is a "method of operation," specifically excluded from copyright under 17 U.S.C. § 102(b). *Id.* at 817, Pet. App. 18a-20a.

The Coalition recognizes that the dividing line between intrinsically functional aspects and copyrightable expression may not always be as clear as we and the First Circuit believe it to be in this case. Nevertheless, we believe that the Court of Appeals' approach is good law – narrow, easy to implement, and likely to lead to consistent determinations regarding the scope of copyright protection for computer program outputs. As such

---

[5] Lotus has acknowledged that there is no claimed similarity between the two programs in "the color, style, layout, or format in which the words or menus are displayed on the screens." Brief for Petitioner, p. 6 n.8.

consistency provides clear roadmarks to secure future software development efforts, it is extremely important to Coalition members, and is in marked contrast to the confusing literal/non-literal analysis followed by the District Court and urged by Lotus and its amici.

It also bears mentioning that the use of software has proliferated into practically every market; thus, if software innovation is chilled in an atmosphere of overbroad copyright protection, there will arise a very clear potential of adversely impacting virtually every aspect of American business.

B. The First Circuit's Decision Does Not Leave the Software Industry With Inadequate Intellectual Property Rights.

Lotus' amici warn that the Court of Appeals' decision will have "a substantial and detrimental effect" on the software industry. Hardware Cos. Brief, p. 17. First, they contend that the appellate court's decision, by unduly restricting the scope of copyright, undermines a system of protection under which the industry has been able "to flourish and grow." *Id.* Second, they suggest that because user interface design "requires a substantial investment in creative expression," the failure to extend copyright protection to such efforts will "undermine [the] incentives to innovate." *Id.* at 18. The Coalition disagrees on both counts.

To begin with, the American software industry has not prospered because of, or even during, a regime of overly broad copyright protection. Indeed, during the crucial formative years of the industry, the industry

looked for guidance to Judge Patrick Higgenbotham's decision in *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003 (N.D. Tex. 1978). In that decision, Judge Higgenbotham denied copyright protection to the plaintiff's format for data input, which the defendants had used in order to make their respective programs compatible without loss of data. *Id.* at 1008. The decision in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1239-40 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987), cast doubt on developers' ability to ensure interoperability; however, those doubts largely were resolved in the early 1990's, when the Second, Ninth and Federal Circuits expressly rejected the extension of copyright to protect those functional elements required to ensure program compatibility, and the Ninth Circuit held that a list of user commands for a computer program was not copyrightable subject matter.[6] Thus, for most of the industry's life, it has grown and prospered in an environment in which the scope of copyright protection was carefully circumscribed and hence fostered the development of more interoperable product innovations.

As for amici's second point, in language disturbingly reminiscent of the "sweat of the brow" theory, their argument presupposes that if copyright protection is not extended to functional aspects of output such as menu

---

[6] *See Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 709-710 (2d Cir. 1992); *Sega Enterprises, Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1993); *Atari Games Corp. v. Nintendo of America, Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *Ashton-Tate v. Ross*, 916 F.2d 516, 521 (9th Cir. 1990).

trees, the developer will not be able to obtain any intellectual property rights in the work. However, denying copyright protection to Lotus' menu command hierarchy does nothing toward denying such protection to other elements rightfully protected by copyright. Further, the Lotus amici have also seemingly forgotten another primary source of federal intellectual property right protection – the patent laws. (Lotus' parent, IBM, has not suffered any similar memory lapse, having obtained a patent on a menu command hierarchy. *See* JA 856-863.)

We believe that amici's oversight is significant. If one approaches the issue as if the mere circumscription of copyright protection for functional aspects of program outputs strips those aspects of all protection, the inclination may well be to err on the side of overprotection. This, however, ignores the protections available through the patent system, a far superior vehicle for providing protection for deserving functional aspects of program outputs. The patent claim process allows both the developer and its competitors to know at the outset the boundaries of the developer's property rights; it enables a competitor, through the re-examination process, to test the validity of the developer's rights without having to first produce the competing product and take a chance on being sued; and, by requiring proof that the work be new and not obvious, it protects against unjustified monopolies. As one commentator recently observed, the dynamics of this case would have been quite different under the patent laws:

> If Lotus had been granted a patent on all these aspects of 1-2-3, Borland could have requested re-examination by the PTO for a relatively small

> fee, and would then have known whether or not it could use these aspects of Lotus 1-2-3 in Quattro. . . . If the menus and macro language of 1-2-3 had been patented, rather than copyrighted, it is conceivable that Borland could have designed around the patent claims. At least, Borland would have had very clear notice of the extent of the property rights to which Lotus was entitled. Instead, it was left to a federal district court judge to decide that Lotus was entitled to a copyright in the menus and macro language, even though no copyright for these aspects of the software was ever filed, examined, or granted.[7]

Patent is the proper province for functional elements of a software program. In fact, patents have already issued on menu command hierarchies. *See* JA 842, 856 (selected portions of U.S. Patent Nos. 4,989,141; 4,611,306). If a functional element can withstand the rigors of patent examination, upon the patent's issuance the holder has a contract with the United States for a monopoly reasonably limited in time. To afford copyright protection to such functional elements is to effectively grant protection which is tantamount to patent protection (without the prerequisites for patent protection such as novelty and nonobviousness), but for anywhere from more than three to ten times as long. As is sometimes said, "you can't have it both ways."

At the least, through the patent process Borland would have had very clear notice of the extent of the

---

[7] Nelson R. Capes, *The Software Copyright "Super Patent,"* 11 Computer Lawyer 8,14 (June 1995).

property rights to which Lotus was entitled. Instead, it was left to a federa' district court judge to decide that Lotus was entitled to a copyright in the menus and macro language, even though no filing, examination, and issuance would have occurred as would be the case in the more appropriate patent arena.

We agree with one commentator's conclusion that "*Lotus v. Borland* illustrates the dangers inherent in [copyright's] ex-post, time-of-infringement approach to determining the scope of protection," under which "an early developer can carve out a potentially lucrative realm of protection at very little cost and without notifying competitors of exactly what is being protected." *Id.* at 15. The "lucrative realm," however, would be an unearned windfall if copyright protection were to run to such functional aspects as Lotus' menu command hierarchy. In our view, the First Circuit's circumscription of copyright protection is fair to the software developer, fair to its competitors, and better for the industry as a whole.

Furthermore, the First Circuit decision properly implements the balanced objectives of copyright as detailed in the CONTU Final Report at 12: "[t]o provide reasonable protection for proprietors without unduly burdening users of programs and the general public, the following statements concerning program copyright ought to be true:

(1) Copyright should proscribe the unauthorized copying of these works.

(2) Copyright should in no way inhibit the rightful use of these works.

(3) Copyright should not block the development and dissemination of these works.

(4) Copyright should not grant any more economic power than is necessary to achieve the incentive to create."

IV. CONCLUSION

For all of the foregoing reasons, we ask this Court to affirm the decision of the First Circuit.

Respectfully submitted,

THOMAS F. VILLENEUVE
(Counsel of Record)
GUNDERSON DETTMER STOUGH
VILLENEUVE FRANKLIN &
HACHIGIAN, LLP
600 Hansen Way, Second Floor
Palo Alto, California 94304
(415) 843-0500

Attorneys for Amicus
Software Industry Coalition Joined by Others

December 1995

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

No. 94-2003

IN THE

Supreme Court Of The United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF OF *AMICUS CURIAE* ALTAI, INC. IN SUPPORT OF RESPONDENT

CORRECTED COPY

SUSAN GERTRUDE BRADEN *
INGERSOLL AND BLOCH, CHARTERED
THE SHERMAN HOUSE
1401 SIXTEENTH STREET, N.W.
WASHINGTON, D.C. 20036
(202) 232-1015

Attorney for Amicus Curiae
* *Counsel of Record*

Balmar Legal Publishing Department, Washington, D.C. (202) 682-9800

BEST AVAILABLE COPY

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	ii
INTEREST OF AMICUS CURIAE	1
STATEMENT OF THE ARGUMENT	2
ARGUMENT	3
THE COPYRIGHTABILITY OF A COMPUTER PROGRAM IS AN ISSUE OF SUBJECT MATTER JURISDICTION WHICH SHOULD BE DETERMINED BY THE COURT AS A THRESHOLD ISSUE, WITH THE PLAINTIFF ASSUMING THE BURDEN OF PROOF UNDER SECTION 102(b)	3
CONCLUSION	6

TABLE OF AUTHORITIES

Cases	Page
Bibbers Sys., Inc. v. Colwell Sys., Inc., 893 F.2d 1104 (9th Cir. 1990)	4
Computer Assocs. Int'l., Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992)	2, 6
Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544 (E.D.N.Y. 1991)	1, 6
Statutes and Rules	
17 U.S.C. § 102(a)	3
17 U.S.C. § 102(b)	Passim
17 U.S.C. § 410(c)	5
Sup. Ct. R. 36	1
Fed. R. Evid. 706	6

IN THE

Supreme Court Of The United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

BRIEF OF *AMICUS CURIAE* ALTAI, INC. IN SUPPORT OF RESPONDENT

INTEREST OF AMICUS CURIAE

Altai, Inc. ("Altai") submits this amicus curiae brief, pursuant to Sup. Ct. R. 36, in support of respondent Borland International, Inc. ("Borland").

Altai designs, develops and markets job scheduling software programs for IBM and IBM-type mainframe computers. Job scheduling programs "create a schedule that specifies when the computer should run various tasks, and to control the computer as it progresses through that schedule." *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 552 (E.D.N.Y. 1991) (*"Altai I"*). In August 1995, Altai was acquired by PLATINUM

technology, inc., and now is a wholly owned subsidiary of that company.

Altai has no interest in either of the parties to this dispute or in the outcome of this case if, as the parties before this Court agree, the holding in *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992) (*Altai II*) is not at issue.[1]

We agree with the characterization of amici for petitioner Lotus Development Corporation ("Lotus") that the instant case is "much more than a dispute between the parties: it is a clash between two fundamentally-opposed views of the protection Congress provided for computer software in the Copyright Act." Motion of Amici Digital Equipment Corporation, The Gates Rubber Company, Intel Corporation and Xerox Corporation at 2.

STATEMENT OF THE ARGUMENT

This Court has granted certiorari to determine whether a computer "menu command hierarchy" is uncopyrightable subject matter under Section 102(b) of the Copyright Act as a "method of operation." Altai supports the conclusion reached by the First Circuit. We are concerned, however, that affirmance of the First Circuit's decision, without reference to a procedural and time framework in which copyrightability must

---

[1] The Second Circuit continues to retain jurisdiction over this case. On December 17, 1992, the misappropriation of trade secret claim was remanded to the trial judge. *Altai II*, at 721. On August 20, 1993, that claim was dismissed pursuant to the applicable Texas statute of limitations. *Computer Assoc. Int'l., Inc. v. Altai, Inc.*, 832 F. Supp. 50, 54 (E.D.N.Y. 1993). On April 7, 1994, the Second Circuit certified to the Supreme Court of Texas the issue of whether the discovery rule applies to misappropriation claims and whether the state's open courts provision applies. *Computer Assoc. Int'l., Inc. v. Altai, Inc.*, 22 F. 3d. 32, 37 (2d. Cir 1994). On June 8, 1995, the Supreme Court of Texas ruled in Altai's favor on both issues in a 7-2 decision. *Computer Assoc. Int'l., Inc. v. Altai, Inc.*, No. 94-0433 slip op. (Tex., June 8, 1995), __S.W. 2d__. On June 27, 1995, a petition for rehearing was filed, which is pending before the Texas Supreme Court.

be established, will do little to assist district courts in managing complex computer software cases, which have been characterized by both courts below as "uncharted territory."

The procedural suggestions offered for the Court's consideration would not interfere with the legitimate enforcement of copyright to protect property interests, but would go a long way to end the use of copyright litigation to eliminate or deter competition and innovation.

ARGUMENT

THE COPYRIGHTABILITY OF A COMPUTER PROGRAM IS AN ISSUE OF SUBJECT MATTER JURISDICTION WHICH SHOULD BE DETERMINED BY THE COURT AS A THRESHOLD ISSUE, WITH THE PLAINTIFF ASSUMING THE BURDEN OF PROOF UNDER SECTION 102(b)

Congress empowered and entrusted the courts with responsibility for ascertaining the copyrightability of computer programs. Section 102(a) provides a specific list of works of authorship that Congress determined were the proper subject matter of copyright. Conversely, Section 102(b) provides a catalog of items that Congress specifically excluded from copyright protection.

Whether a particular computer program or any elements thereof are proper subject matter for copyright is an issue of jurisdiction which should be determined by the court as a threshold issue, and the burden of proof should reside with the plaintiff. If a claim of infringement does not concern an interest in legitimately copyrightable subject matter, it should be dismissed forthwith, regardless of any other circumstances, such as whether the plaintiff created a pre-existing computer program that is substantially similar to a competitor's subsequently developed program. Federal courts do not have jurisdiction over

computer software infringement claims, if the programs at issue are not copyrightable under Section 102(b).

Therefore, if copyrightability is disputed, it is incumbent on the district court to require any plaintiff alleging infringement of a computer program to establish and assume the burden of proof as to the statutory elements of Section 102(b) and for the court to make findings at the onset of the litigation. In particular, this should be done *before* allowing the plaintiff and its counsel to use the liberal discovery afforded under the Federal Rules of Civil Procedure to establish copying and/or peruse a rival's competing computer program(s) and ascertain otherwise confidential business and sales information to determine "damages."[2]

Today, merely alleging copyright infringement of computer software provides the plaintiff with license to impose significant market and litigation costs, which can eliminate competition and create entry barriers that far exceed the very limited scope of the monopoly Congress intended to convey by copyright.[3] Copyright infringement litigation is a powerful weapon in the battle for consumer dollars and market share in the computer software industry. A defendant's business can literally be destroyed by the uncertainty and doubt cast on its products by the mere filing of a complaint. Years later, if and when no infringement is found, no remedy is available for the damage caused to a business merely by the pendency of the suit.

Yet the First Circuit cites with approval *Bibbers Sys., Inc. v. Colwell Sys., Inc.*, 893 F.2d 1104, 1106 (9th Cir. 1990), which held that:

---

[2] In that regard, we suggest that the district court in the instant action abused its discretion in not certifying its initial decision for appeal. Borland Certiorari Opposition at 11-12.

[3] The adverse market effect of the instant litigation on Borland is a matter of public record. Borland Certiorari Opposition at 12-13.

> A certificate of copyright registration constitutes *prima facie* evidence of copyrightability and shifts the burden to the defendant to demonstrate why the copyright is not valid.

But Section 410(c) of the Copyright Act is more limiting. It provides only that:

> In any judicial proceedings the certificate of registration made before or within five years after first publication of the work shall constitute prima facie evidence of the validity of the copyright and of the facts stated in the certificate. The evidentiary weight to be accorded the certificate of registration shall be within the discretion of the court.

17 U.S.C. § 410(c) (emphasis added).

Where the copyrightability of a computer program is disputed, registration alone should not satisfy the plaintiff's burden of proof on that issue, particularly in cases where non-literal aspects of a computer program are at issue. Because of the inherently utilitarian nature of computer programs, this Court should exercise its supervisory authority to instruct the lower courts to require a plaintiff to assume the burden of proof to establish that the particular program or elements thereof at issue are subject matter that is not excluded by Section 102(b). Since, in most instances, this will entail the testimony of technical experts in the field of computer science, the defendant should be allowed not only to cross examine the plaintiff's experts, but also to proffer rebutting testimony through its own experts. In cases of computer software infringement, the assistance of a court appointed expert may be particularly helpful to the district court, at this stage in the litigation, in evaluating the quality and context of technical evidence adduced to satisfy Section 102(b) and in creating a robust factual record from which that court, and any subsequent appellate court, may determine whether the plaintiff's burden as to copyrightability has been met — *i.e.*, to

establish that its computer program is *not* an idea, procedure, process, system, method of operation, concept, or a discovery.[4]

We believe that had the district court required Lotus to assume the burden of proof as to Section 102(b), Lotus's menu command hierarchy would have been found to be uncopyrightable.

CONCLUSION

The judgment of the court below should be affirmed.

Respectfully submitted,

SUSAN GERTRUDE BRADEN *
INGERSOLL AND BLOCH, CHARTERED
THE SHERMAN HOUSE
1401 SIXTEENTH STREET, N.W.
WASHINGTON, D.C. 20036
(202) 232-1015

Attorney for Amicus Curiae
* Counsel of Record

December 8, 1995

---

[4] An important contribution to the analysis of the Second Circuit in *Altai II* was the decision of former U.S. Circuit Court Judge George C. Pratt, sitting by designation as the district court, to appoint an expert, pursuant to Fed. R. Evid. 706 to advise the court as to the technical properties of the computer programs at issue in that case. *Altai I*, at 549. With an independent reference and source of technical advice, both the trial and appellate courts were better equipped to evaluate the computer programs at issue and the quality and credibility of the parties' experts and the opinions they advanced.

The district court in the instant case did not have the benefit of such advice and was left to ferret out technical matters, which exceed the expertise of most federal court judges and counsel.

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

No. 94-2003

In The

Supreme Court of the United States

October Term, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL, INC.,

Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

BRIEF AMICUS CURIAE OF USERS GROUPS IN SUPPORT OF RESPONDENT

REX S. HEINKE
(Counsel of Record)
GIBSON, DUNN & CRUTCHER
333 South Grand Avenue
Los Angeles, CA 90071
(213) 229-7000

JONATHAN C. DICKEY
GIBSON, DUNN & CRUTCHER
3000 Sand Hill Road
Building 1, Suite 285
Menlo Park, CA 94025
(415) 926-7300

Attorneys for Amici
Listed in Appendix A

December 1995

COCKLE LAW BRIEF PRINTING CO., (800) 225-6964
OR CALL COLLECT (402) 342-2831

TABLE OF CONTENTS

Page

INTEREST OF THE AMICI CURIAE 1

SUMMARY OF THE ARGUMENT 2

ARGUMENT 5

I. Macro Programs are Integral to the Use of Personal Computers in the Workplace..... 5

II. Users Have Made An Enormous Investment in Learning the Command Structure and Devising Macros for Lotus 1-2-3 8

A. The Users' Skill Set in Lotus 1-2-3..... 8

B. The Users' Acquisition of 1-2-3 Macros.. 14

III. Lotus' Monopolization of the 1-2-3 Command Hierarchy Would Inflict Substantial Harm on Consumers and Would Encourage Stagnation in the Computer Software Market..... 15

A. Consumers Would Become Locked-In To Lotus 1-2-3..... 16

B. Consumers Would Lose Their Intellectual Property Rights in Their Own Macros... 20

CONCLUSION 23

DESCRIPTION OF AMICI CURIAE USERS GROUPS .....App. 1

TABLE OF AUTHORITIES

Page(s)

CASES

Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340 (1991) 3

Fogerty v. Fantasy, Inc., ___ U.S. ___, 114 S.Ct. 1023, 127 L.Ed.2d 455 (1994) 1

Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992) 6, 9, 15

Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993) 6

Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) 16

PUBLICATIONS

Caren Goldman, *Traditional Jobs Replaced by New Opportunities*, PLAIN DEALER (Cleveland, Ohio), Oct. 4, 1994, at 21 ..... 13n

Jennifer King, *How State of the Art Technology Transformed Sears Roebuck & Co.'s Law Department*, CORPORATE LEGAL TIMES, Aug. 1992, at 6 ..... 13n

Woody Leonhard, et al., *Hacker's Guide to Word for Windows* (1995) 8, 8n

Lotus 1-2-3 Reference Manual, Release 2.0 ..... 9n

E.M. Lunsford, *Classic 1-2-3 Macros* (3d ed. 1994) 10

Steve Stecklow, *1-2-3 Training Options: Instructions for Lotus Development Corp.'s Lotus 1-2-3 Spreadsheet Software and Other Software Packages*, ASAP: LOTUS, July 1992 at 36 ..... 13n

Mary C. Thorsby, *Levi's Delivers Lessons in Learning*, MISS. BUS. J., Oct. 18, 1993 at 7 ..... 13n

INTEREST OF THE AMICI CURIAE

The *amici curiae* Users Groups consist of 61 personal computer ("PC") users groups, with a total membership of approximately 53,000 corporate and individual users. The Users Groups range in size from 50 to over 15,000 members, and include among their members many Fortune 500 businesses and other firms with multiple offices around the country. The Users Groups generally have been organized to provide members with a forum for the exchange of information about computer software and hardware, and to help educate group members and the general public about the development and use of personal computer products. A description of each of the participating Users Groups is set forth in Appendix A.[2] The Users Groups have received the consent of the attorneys for both Petitioner and Respondent to file this *amicus* brief.

As this Court has observed, the federal copyright laws, consistent with their constitutional mandate, "must ultimately serve the public good." *Fogerty v. Fantasy, Inc.*, ___ U.S. ___, 114 S.Ct. 1023, 1029, 127 L.Ed.2d 455 (1994). In this case, the Users Groups believe that, to a large extent, they represent personal computer software consumers – in other words, the "public" for whose benefit the copyright laws operate. The Users Groups are well-positioned to provide the Court with important perspective regarding the investment made by users in the Lotus 1-2-3 menu language, and the harm that consumers would suffer if copyright protection is extended to menu command hierarchies, such as the Lotus 1-2-3 menu tree.

---

[2] This brief was approved pursuant to the balloting procedures of the signatory users groups who exercised complete control over its editorial contents. Borland helped to defray the costs of printing.

Although our position in this brief may benefit respondent Borland International, Inc., ("Borland"), the Users Groups have no allegiance to any particular product or computer vendor. Members of the Users Groups have purchased products by both Borland and petitioner Lotus Development Corporation ("Lotus"), as well as the software products of hundreds of other software and hardware computer vendors, including Microsoft and IBM.

SUMMARY OF THE ARGUMENT

Rarely has an intellectual property case before this Court presented such a stark threat to the future development of one of this country's key industries, or invited such widespread legal analysis and commentary over the proper role of copyright in our nation's tradition of technological innovation. Since this litigation first commenced five and a half years ago, evolving standards of intellectual property protection have challenged the high technology industry, and the legal community, as in no other time in this country's history. Underlying this case is the fundamental issue – debated every day in the boardroom and in the courtroom – of where invention leaves off and appropriation begins. This litigation now places this Court squarely at the crossroads of this debate, and necessitates the setting of a course by this Court which will promote the continued development and success of this country's computer software industry in an era marked by rapid and accelerating technological change.

The *amicus curiae* Users Groups respectfully submit that this case presents a unique and timely opportunity to clarify the role of copyright in this developing industry, and to re-affirm that the role of our copyright laws is to serve the *public interest*. In this amicus brief, the Users Groups hope to educate this Court on the critical importance of the Lotus 1-2-3 menu command hierarchy to users of spreadsheet applications, and on how the Lotus copyright claim in this case poses a substantial economic and technological threat to computer users.

From its beginning, federal copyright law has been based on a careful balance between providing sufficient protection to encourage authors to create new works, and "encourag[ing] others to build freely upon the ideas and information conveyed by a work." *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 350 (1991). In attempting to extend copyright protection to the 1-2-3 menu command hierarchy, the District Court failed to strike the proper balance. Instead, the District Court's decision to protect the 1-2-3 command hierarchy through copyright would have seriously harmed consumers and deterred the production of new, innovative products, while allowing Lotus to unfairly reap profits from value that has been added to 1-2-3 by *users, not* by Lotus.

The Users Groups believe that this imbalance may have resulted, at least in part, from the District Court's failure to recognize the extent to which users of computer programs invest in and rely upon the 1-2-3 command hierarchy. As described in more detail below, this investment by consumers has occurred at two levels. First, customers have invested an enormous amount of time and resources in learning and becoming proficient in the

1-2-3 command language. Second, they have created or purchased "macro" programs, often highly complex and sophisticated, that depend upon the 1-2-3 menu tree in order to run. For many corporate and other business users, these macro programs have become an indispensable part of how they run their business.

If Lotus is granted a monopoly in the 1-2-3 menu, users would be unable to switch to a new, superior spreadsheet product without losing this enormous investment. Very much like learning a different arrangement of keys on a typewriter or computer keyboard, users would have to learn an entirely new set of keystroke sequences. Moreover, unlike switching keyboards, users here would have to replace at great expense the macro programs that they created using 1-2-3. Finally, to avoid becoming even more deeply locked-in in the future, existing 1-2-3 consumers would be less likely to create or program new macros, thus deterring the creation of valuable new works – the very aim of copyright protection. Thus, consumers would be harmed twice over: they would be captive to 1-2-3 even if a superior product were developed, and they would be deterred from creating valuable new works lest the technological tie that binds them to Lotus 1-2-3 be tightened even further.

Given the enormous number of users who have purchased or created 1-2-3 macros and made them an integral part of their business, the harm to consumers from Lotus' proposed rule would be substantial even if command languages enabling the creation of macros were characteristic only of spreadsheet products. As explained in more detail below, however, macro languages are a

pervasive and central feature in the effective use of personal computers in the American workplace. Allowing the dominant makers of a handful of operating system and application software programs to monopolize the macros that run on them would dramatically reduce the degree of innovation and productivity that users can achieve through personal computing.

ARGUMENT

The following argument is divided into three sections. The first section explains briefly how macro programs currently are used, and provides some examples of popular programs other than 1-2-3 that are used as macro platforms. The second section explains in more detail the nature and scope of the investment made by users in learning the 1-2-3 menu command language and in creating 1-2-3 macros. The final section describes more fully the harm to the public that would result from allowing Lotus to reap where it has not sown.

I. Macro Programs are Integral to the Use of Personal Computers in the Workplace.

Macro programs are a series of recorded keystrokes which may be run with a single command. By allowing the automation of complex or repetitive command sequences, macros permit users, in effect, to develop their own application programs. Thus, for example, macros may be used to transform generic database programs into order entry systems for small businesses, timekeeping

tools for professionals, and patient record repositories for hospitals.

In its earlier opinions, the District Court described certain very simple 1-2-3 macros, most of which consisted of only two or three menu commands. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) ("*Borland II*"), Pet. App.[3] 110a-112a, 123a; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223 (D. Mass. 1993) ("*Borland IV*"), Pet. App. 31a-33a. These simple examples may aid in understanding how macros work, but they do not give a proper sense of how macro programs, many of which consist of thousands of lines of commands, actually are used in the workplace. For example, in a commercial law firm, macros written for word processing software enable individual users throughout the firm to format a document as a legal pleading, a letter, or an expense voucher in a matter of seconds. At the same firm, macros written for telecommunications programs automate the complex series of commands required to connect computer users to electronic bulletin boards, the Internet, and services such as Lexis and Westlaw.

Because of the enormous power and flexibility that macros provide, a majority of the most popular operating systems and applications platforms – spreadsheets, word processors, database programs, and telecommunications programs – allow users to create macros that automate the commands found in the program's menu hierarchy.

---

[3] "Pet. App." refers to the Appendix to Petition for Writ of Certiorari. "JA" refers to the Joint Appendix.

Examples of flagship products with macro-creation facilities include Novell's WordPerfect and Microsoft's Word (word processing), Microsoft's Windows (operating system), Lotus' Notes (groupware), Microsoft's Excel (spreadsheet), and Borland's Paradox (database).

Corporate users often will employ different spreadsheet or other applications programs within the same company. For example, a company's marketing department might run one spreadsheet program, while the finance department runs another; an acquired subsidiary might run a different program from other subsidiaries; or an individual user might prefer one program for graphic displays and another for numeric calculations. In order to utilize the work generated using these diverse programs by different departments, corporations create macros that collect, perform calculations upon, and form data from these different departments. For example, a firm might create an extensive and complex macro program to assist in the annual budgeting process. The macro could be used to obtain sales estimates from the marketing department, manufacturing capacity data from operations, and staffing numbers from human resources. This data would, in turn, be formatted automatically by the macro into a final document for presentation to management. In order effectively to use such valuable, time-saving macro programs, however, the spreadsheet program used by each department must be macro-compatible – *i.e.*, capable of running macros written for – the spreadsheet programs of the other departments. Indeed, in the absence of macro compatibility, companies would be more likely to standardize on a single spreadsheet product, even if it is less

than optimal for many purposes, simply to allow uniform access to and processing of valuable corporate data.

The demand for macros has spawned an entire industry, inspiring third-party authors to create guidebooks and manuals that instruct consumers how to use the menu structure and devise macros using the menu language of numerous application programs. For example, the 800-page *Hacker's Guide to Word for Windows*[4] describes the macro capabilities of Microsoft's Word word processing software. The convenience of macros even has led to the development of programs, such as QuicKeys for the Apple Macintosh, the sole function of which is to permit users to create their own macros. With the development of such products, the pervasiveness of macros will only increase in the future.

II. Users Have Made An Enormous Investment in Learning the Command Structure and Devising Macros for Lotus 1-2-3.

Users invest in the menu command hierarchy of a program such as 1-2-3 in two principal ways: first, in the time and effort that they devote to becoming proficient in the command hierarchy; and second, in the purchase or creation of macros that depend upon the command hierarchy for their proper operation.

A. The Users' Skill Set in Lotus 1-2-3

In *Borland II*, the District Court acknowledged that the menu command hierarchy "is a fundamental part of

---

[4] Woody Leonhard, et al., *Hacker's Guide to Word for Windows* (1995).

the functionality of [both the 1-2-3] keystroke sequences and the macro language." 799 F.Supp. at 207, Pet. App. 110a. With respect to the first of these uses, an example given by the District Court helps to illustrate how keystroke sequences serve as the equivalent to individual keys on a keyboard. To obtain a number that is formatted in dollars, a user would strike four keys: \RFC. The "\" initiates a command sequence, the character "R" implements the "Range" command, the character "F" implements the "Format" command, and the character "C" implements the "Currency" command. *Id.* The "\RFC" sequence thus serves the same kind of role that the "$" key serves on a QWERTY keyboard. Another illustrative example is the keystroke sequence used to delete the contents of a particular cell on the spreadsheet. The user does not hit the "delete" key on the keyboard. Instead, the user types the sequence "\RE," for "command," "Range," "Erase."[5]

The other role served by the menu command hierarchy is to act in effect as the programming language for macro programs. Just as when programming in BASIC or C, a macro program developer is concerned primarily with the function that each keystroke or word will

---

[5] Lotus complains that the Court of Appeals' analogy to buttons on a Video cassette recorder ("VCR") is inaccurate because a 1-2-3 user cannot perform an operation by hitting a single key. *See* Brief for the Petitioner at 9 n.15. This is a distinction without a difference: numerous 1-2-3 operations, for example, consist of 3-keystroke sequences, such as saving a worksheet ("\FS") and erasing a worksheet ("\WE"). *See Lotus 1-2-3 Reference Manual, Release 2.0* at 91, 57. It also suggests that most functions on a typical VCR can be accomplished with a single button – an assumption that does not coincide with our experience with such machines.

invoke. Third party authors thus have expressly equated the 1-2-3 command structure with a programming language. *See, e.g.*, E.M. Lunsford, *Classic 1-2-3 Macros*, p. 3 (3d ed. 1994) In fact, according to Lunsford, "some studies have shown that more people are programming with macros than with any other language." *Id.*

Users have expended countless hours, at significant cost, learning how to use the command hierarchy and to create 1-2-3 macros. As the examples given above demonstrate, the proper keystroke sequences are not necessarily intuitive: for example, saving a worksheet ("\FS" or "File Save") and erasing a worksheet ("\WE," or "Worksheet Erase") are not even on the same fork of the 1-2-3 menu "tree." A user must spend a great amount of time learning the menu structure in order to be able to operate the program in a "heads down" fashion. To an extent not true of an ordinary keyboard, and contrary to the suggestion in Lotus' Brief (Brief for the Petitioner at 11), "hunt and peck" simply is not a viable option. Indeed, if the user had to look up the difference between "\WGC" (setting column width for the entire spreadsheet), "\WCS" (setting width for a particular column) and "\WGL" (aligning column labels) every time the user wanted to readjust the columns, the spreadsheet might well never get finished.

The complexity of the keystroke sequences is itself an important reason why users have invested enormous resources in learning how to program macro "shortcuts." Thus, for example, 1-2-3 requires a user to type up to 15 keystrokes in order to enter the date – that is, unless the user creates a date macro. It is perhaps not surprising under these circumstances (and given Lotus' historic

dominance of the spreadsheet market),[6] that colleges and universities around the country provide training on how to use and create macro programs in 1-2-3. For instance, the Users Groups pointed out to the Court of Appeals that in the surrounding Foston area, all undergraduate business students at Northeastern University must take a four credit hour course entitled "Computer-Based Information Systems." A significant amount of the course is devoted to learning how to use and program macros in Lotus 1-2-3. Similarly, all undergraduates at Babson College in Babson Park, Massachusetts are required to take a four credit hour course entitled "Introduction to Information Systems." Approximately one-third of the course is devoted to learning Lotus 1-2-3.

---

[6] In the Court of Appeals, Borland cited media sources as reporting in August 1992 that Lotus claimed market share of approximately 70%. Brief of Defendant/Appellant Borland International, Inc. at 21 (1st Cir.) (No. 93-2214). Lotus suggests that the Court (and presumably consumers) should draw solace from the fact that Microsoft was able to use its monopoly in the operating systems market to replace Lotus as the dominant spreadsheet provider. Brief for the Petitioner at 16. The Users Groups draw scant comfort from this example. To begin with, Lotus never sought to prevent Microsoft from making its Excel program compatible with 1-2-3 macros, so Microsoft was able to "migrate" 1-2-3 users to Excel. More fundamentally, it illustrates the extent of the dependence of programs on the software platform underlying it. Thus, just as 1-2-3 macros depend on their compatibility with the underlying spreadsheet program, 1-2-3 depends on its compatibility with the underlying operating system. Because Microsoft, through its control of the operating system socket, was able to ensure that Excel was the first Windows-compatible spreadsheet, it was able to cut the ground out from under 1-2-3.

Responding to user demand, continuing education divisions of many universities also offer such courses. For instance, Harvard University's Office of Information Technology offers two courses in 1-2-3 training, one lasting seven and the other fourteen hours. Within close proximity to this Court, Georgetown University and the University of Virginia both offer classes featuring Lotus 1-2-3. Georgetown offers a 12-hour course on "Lotus Fundamentals," while UVA offers Lotus training in two 3-unit courses entitled "Microcomputer Applications in Computer Information Systems," and "Microcomputer Applications for Managers."[7]

In addition to university-sponsored coursework in Lotus 1-2-3 macros, a number of companies regularly provide training to U.S. government agencies and the Department of Defense, as well as to companies doing business with such government agencies. For example, ExecuTrain provides a course on "Lotus 1-2-3 for Windows Advanced Spreadsheet and Database" for technology companies such as IBM (now the parent company of Lotus), Intel, Digital Equipment Corporation, Hewlett-Packard, Honeywell, and Sony, and for Fortune 500 companies such as Dow Chemical, Eastman Kodak, General Electric, General Motors, Chrysler and AT&T. Similarly, Future Enterprises, Inc. provides a course on "Advanced Spreadsheet Concepts Using Lotus 1-2-3" for such government agencies as the Department of Justice, the Internal Revenue Service, the Department of Energy, the

---

[7] Similarly, the University of Maryland Business School offers a course called "Business Information Systems," for which training in Lotus 1-2-3 is a prerequisite.

National Labor Relations Board, and the Central Intelligence Agency. Both ExecuTrain and Future Enterprises are authorized Lotus training centers.[8]

Finally, private employers have committed enormous resources to training employees in the use of Lotus 1-2-3, and 1-2-3 macros. Spreadsheets are integral to the efficient and effective operation of virtually every major segment of business, from marketing to manufacturing, and employers accordingly have included 1-2-3 training as part of basic orientation for many thousands of their employees. Firms that have provided 1-2-3 training to employees include Levi Strauss & Co.,[9] Sears Roebuck & Co.,[10] New York Telephone,[11] and Area Temps.[12] Similar examples could be found many times over throughout the country, with many other types of computer programs that employ a complex menu structure and permit the creation of labor-saving macros.

---

[8] The U.S. Department of Agriculture operates its own training center, with a course entitled "Lotus 1-2-3 for Windows: Intermediate," available to all federal government agencies and the Department of Defense.

[9] Mary C. Thorsby, *Levi's Delivers Lessons in Learning*, Miss. Bus. J., Oct. 18, 1993 at 7.

[10] Jennifer King, *How State of the Art Technology Transformed Sears Roebuck & Co.'s Law Department*, Corporate Legal Times, Aug. 1992, at 6 (legal department trained to use 1-2-3).

[11] Steve Stecklow, *1-2-3 Training Options; Instructions for Lotus Development Corp.'s Lotus 1-2-3 Spreadsheet Software and Other Software Packages*, ASAP: Lotus, July 1992, at 36.

[12] Caren Goldman, *Traditional Jobs Replaced by New Opportunities*, Plain Dealer (Cleveland Ohio), Oct. 4, 1994, at 21 (small midwestern temporary employment agency which also offers 1-2-3 training to employers).

B. The Users' Acquisition of 1-2-3 Macros

Users not only have an investment of time spent learning 1-2-3, but they also have an investment in the macros themselves. Some of this investment is monetary, inasmuch as 1-2-3 macros are available in virtually any computer store. For instance, "101 Macros Plus for Lotus 1-2-3" offers approximately 150 ready-to-use macros that allow users to create graphic slide shows and to automate their spreadsheets. Another product named "@analyst" provides financial analysis for various securities, including bonds, bills, certificates of deposit, options, and mortgage-backed securities. Likewise, "Crystal Ball," which aids financial analysts and other professionals, is a forecasting and risk-analysis software package based upon Lotus 1-2-3 for Windows.

Perhaps more typical, at least for large users, is the creation of the user's own proprietary 1-2-3 macros. For example, large accounting firms have developed macros which download information stored on mainframe computers into a spreadsheet that can be used by individual auditors working on personal computers anywhere within the firm. Similarly, stock brokerage firms have developed proprietary macros that enable individual brokers to perform sophisticated risk analyses, while commercial brokerage firms run 1-2-3 macros in order to enable their employees to perform rapid and sophisticated analyses of their listings. Not surprisingly, Lotus has actively encouraged the development of customized macro programs, recognizing that such macros greatly

increase the range of uses to which 1-2-3 can be put (and hence the size of the market of potential customers).[13]

Common to all 1-2-3 macros, whether purchased commercially or developed by the user itself, is their dependence upon the 1-2-3 menu commands and their order and sequence (in other words, the menu command hierarchy). The District Court recognized this dependence, finding that "the menu commands are an important part of the functionality of the macros." *Borland II*, 799 F.Supp. at 208, Pet. App. 112a. Put differently, without the ability to use the menu command hierarchy, the macro programs are worthless, and users' investments in those macros are lost.

III. Lotus' Monopolization of the 1-2-3 Command Hierarchy Would Inflict Substantial Harm on Consumers and Would Encourage Stagnation in the Computer Software Market.

The harm that consumers would suffer if Lotus were able to monopolize the 1-2-3 menu hierarchy comes both in the form of time lost spent learning the menu and devising macros that use it, and in the impairment of users' property rights in the macros that they have created or purchased. In both instances, this harm is matched by a concomitant chilling of the incentive of users and third parties to create new and innovative software products.

---

[13] Commencing with the very first 1-2-3 users' manual, Lotus has pointed out the advantages of creating macro programs and has encouraged their use. Declaration of Vern L. Raburn at ¶ 17 and Exh. B, JA 529-530, 535.

A. Consumers Would Become Locked-In To Lotus 1-2-3.

In his concurring opinion in the Court of Appeals, Judge Boudin fairly summarized the threat posed to computer users if Lotus' copyrights for the 1-2-3 menu command hierarchy are sustained by this Court: "the present case is an unattractive one for copyright of the menu. The menu commands (*e.g.*, "print," "quit") are largely for standard procedures that *Lotus did not invent* and are common words that *Lotus cannot monopolize.*" *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 821 (1st Cir. 1995) (emphasis added), Pet. App. 26a. Judge Boudin correctly noted that the importance of the 1-2-3 menu over time "may come to reside more in the investment that has been made by users in learning the menu and in building their own mini-programs – macros – in reliance upon the menu," *Id.* at 819, Pet. App. 24a, and analogized the "lock-in" effect on 1-2-3 users to typists who have become familiar with the QWERTY keyboard. *Id.* at 821, Pet. App. 26a. As that analogy reflects, the barrier is not an insuperable one: a corporation could require all of its typists to switch to a nonstandard keyboard, even though it meant retraining hundreds or thousands of employees. The question simply is how much time already has been invested in the present system, and how much time would be required for retraining.

In the case of the 1-2-3 menu command hierarchy, the answer on both counts is "a lot."[14] For a typical corporate

---

[14] Lotus' argument that there was no evidence in the record regarding the extent of "lock-in" is baffling, inasmuch as it

user, the investment of employee time in the 1-2-3 command structure is both broad and deep. Many, if not most, employees are trained how to use the menu command hierarchy and the macros that routinely are run on the 1-2-3 platform. A few employees or independent contractors may have devoted many thousands of hours to developing proprietary macro programs using the 1-2-3 command language. To switch to an incompatible spreadsheet program, all of that time would have to be spent again, rewriting spreadsheet macros and retraining employees in their use. Moreover, even that investment might not be enough, because, unlike the user of the QWERTY keyboard, the 1-2-3 consumer has the additional hurdle of potentially needing to transfer enormous quantities of data from the old to the new spreadsheet program. The programming entailed in effecting such a transfer can itself be a Herculean task.

The First Circuit decision in this case correctly recognized the importance to computer users of program compatibility, and characterized as "absurd" the proposition

---

appears to the Users Groups to have been a point on which all sides agreed. For example, Lotus' expert stated in his declaration:

> Once users did learn the 1-2-3 menu commands and came to identify them with Lotus 1-2-3 those particular commands and their arrangement clearly acquired great significance. Their *current* value, to both Lotus and Borland, lies precisely in the familiarity that millions of 1-2-3 users have acquired with them in the more than eight years subsequent to the product's introduction.

Declaration of James C. Emery ("Emery Decl.") at ¶ 113, JA 343-344 (original emphasis).

that a computer user should be forced to learn how to perform the same operation in several different ways if the user happens to be running several different programs. This absurd result is precisely what Lotus' copyright theory guarantees – *unless* users simply decide not to purchase or use other programs, and thus become "locked-in" to Lotus. Either way, competition is harmed and computer users will be *powerfully discouraged* from using computer software to create new and useful ways of more efficiently running their businesses and promoting more cost-effective solutions to complex computing problems.[15]

The number of consumers who face the risk of becoming captive to Lotus through their investment in

[15] Lotus entirely misconstrues the competitive issue at stake in this case as one involving the "forfeiting" of copyright protection when a software program "has become – for the moment – the most popular." Brief for the Petitioner at 47. Of course, the First Circuit's opinion finding the Lotus 1-2-3 menu command hierarchy to constitute a "method of operation" under 17 U.S.C. § 102(b) does not depend on any finding that Lotus 1-2-3 is "the most popular" spreadsheet program. One can imagine an equally pernicious competitive impact if purely anonymous computer programmers were permitted to copyright other basic means of computing, and lay claim to all "derivative works" using substantially similar means. The past ten years of patent case law has taught us that the biggest manufacturers do not have a monopoly on obtaining software patents, even with the far more stringent requirements for obtaining patent rights. We can reasonably predict that if Lotus' position is accepted by this Court, a veritable cottage industry will develop involving copyrights to the most basic means of operating computer programs, with disastrous competitive consequences.

the 1-2-3 menu is very large: the Users Groups believe that the number of affected 1-2-3 users likely runs in the *millions*. The harm to consumers, however, would not be limited to 1-2-3 customers. As noted previously, many other software manufacturers – including some of the dominant players in the industry such as Microsoft – have enabled and encouraged consumers to develop their own proprietary macros using menu command languages, and many of these programs have become an indispensable part of their users' efficient operations. Users of these other programs would therefore appear to have the same potential for being "locked in" as those who have allowed their business to become integrally dependent upon 1-2-3.

Given the obvious importance of the "lock in" effect if copyright protection is granted to Lotus in this case, the Users Groups are mystified by Lotus' apparent argument that the issue of "lock-in" due to computer users' need for macro compatibility "was not before the court of appeals and could not properly form a basis for its decision." Brief for the Petitioner at 15. Lotus itself cites to numerous examples in the Record demonstrating the importance of macro compatibility, including the testimony of Lotus' own expert. *Id.* at 13-15. We submit that Lotus' attempt to have this Court simply sidestep this critical issue is not supportable, and Lotus' characterization of the Users Groups' arguments below as "grossly exaggerated" is disingenuous at best.[16] In the final

[16] In the proceedings before the First Circuit, a group of nineteen users groups filed an *amici curiae* brief. Virtually all of those users groups also join in the current brief.

analysis, the very thing that Lotus says was not properly before the First Circuit – macro compatibility – is at the very heart of Lotus' claim of copying, as witnessed by Lotus' own description of "the nature and purpose of Borland's copying." Brief of the Petitioner at 11-15. If, as Lotus posits, the desire to promote macro compatibility is one of the primary motivations for Borland's creation of the "1-2-3 emulation" menu tree, the Users Groups not only find that motive understandable, we applaud it and support it.[17]

B. Consumers Would Lose Their Intellectual Property Rights in Their Own Macros.

The discussion above noted one way in which the QWERTY analogy, though apt, may understate the extent of consumer harm that would result from Lotus' proposed monopoly. There is, in addition, another way in which the analogy should be modified to make it parallel the experience of users of 1-2-3. Customers who have built up a set of 1-2-3 macros not only lose the money they expended in purchasing commercial macros, and the time spent in developing proprietary macros. They also

---

[17] Lotus belittles the Users Groups' competitive concerns by calling them "overstated," and suggesting that consumers are free to pursue antitrust remedies to redress the competitive harm caused by its conduct. However, as the many individual users represented by the Users Groups on this appeal will attest, the option of spending many years and millions of dollars fighting with Lotus and IBM over the competitive harm caused by their unfair copyright monopoly is no option at all for these individuals, or for the non-profit Users Groups who represent their interests here.

lose the separate property rights that they have as the creators of their own macros programs. In other words, it is as though the company that wished to switch from QWERTY to another keyboard had to throw away some of the documents created using the QWERTY keys.

The District Court reserved the question whether 1-2-3 owners actually own the programs that they create using the 1-2-3 macros language, but Lotus and the Users Groups both agree that the end user owns its own macros. Indeed, Lotus confirmed the user's ownership rights in macros during a telephone conference with approximately 16 user group representatives (including several of the undersigned *amici*) on September 1, 1993, shortly before the briefing on appeal in this case. Lotus recorded this conversation, had it transcribed, and posted the transcript to an electronic bulletin board that was available to the public. The following excerpts from the transcript, although not part of the official record in this case, are telling:

> Statement from Mr. Frank Ingari, Lotus Vice President of Marketing: "Customers develop macros. So, there's customer intellectual property involved. We understand that . . . And it is true – customers have rights too. You created intellectual property when you wrote a macro."
>
> Statement from Mr. Tom Lemberg, Lotus Vice President and General Counsel: "Well, I think that, you know, in order for the user – the user, of course, own his or her own macros."

As the owners of these software programs, we submit that users have the right to fully use, transfer, translate, and exploit the product. In preventing a competitor of

Lotus from incorporating a menu command hierarchy essential to the execution of that program, however, Lotus' proposed rule would substantially limit users' intellectual property rights in their own programs.[18]

The mirror image of this restriction on users' rights is the deterrent effect that Lotus' proposed rule would have on the development of programs to compete with 1-2-3. From a user's perspective, Borland engaged in precisely the type of conduct that the intellectual property laws should promote: it provided users with a completely different menu command hierarchy, and at the same time, provided the equivalent of a "socket" to allow users to run their existing macro programs. If construed to prevent competitors offering such compatibility, the copyright law perversely would deepen users' "lock-in" to their existing platform, and would correspondingly reduce the incentive for competitors to try to break the lock.

In sum, there is no public benefit, and much public harm, from allowing Lotus to obtain a monopoly windfall

---

[18] During proceedings in the District Court, Lotus conceded that it is attempting to appropriate value that was created by Lotus 1-2-3 users. Indeed, as Lotus' expert, James Emery, stated, there are a "virtually infinite variety" of ways in which a menu command hierarchy could be structured, and there is nothing particularly special about Lotus' arrangement: in his words, "it would be very difficult to say one [menu arrangement] is better than another." Deposition of James Emery at 202, JA 598. The value came instead from 1-2-3 users: as noted above, he recognized that the 1-2-3 commands' "*current* value, to both Lotus and Borland, lies precisely in the familiarity that millions of 1-2-3 users have acquired with them in the more than eight years subsequent to the product's introduction." Emery Decl. at ¶ 113, JA 344 (original emphasis).

as a result of th value created not by Lotus but by the 1-2-3 users who are members of these Users Groups, and by other 1-2-3 users around the country. The Users Groups therefore respectfully submit that the decision of the Court of Appeals should be affirmed.

CONCLUSION

The Record before this Court fully supports the Court of Appeals' decision in this case that the Lotus 1-2-3 menu command hierarchy constitutes a non-copyrightable method of operation. Any contrary result will lead to an unwarranted and unjustified extension of this country's copyright laws, and a dangerous grant of monopoly power to Lotus and other companies in the rapidly developing computer software industry. At stake are the rights of millions of computer users who have invested enormous resources in their computer software, and who will be the biggest losers if this Court does not affirm. We urge this Court to give due regard in this case to the underlying purposes of our copyright laws, which are to serve the public interest. Otherwise, we fear that copyright protection will serve only the interests of those who, like Lotus in this case, wish to limit further development of the computer software industry to only those few who admittedly helped give birth to the industry, but who now are threatened by a new generation of products

and ideas. Surely, Congress did not intend the copyright laws to reward the past, while risking our future.

Dated: December, 1995

Respectfully submitted,

REX S. HEINKE
(Counsel of Record)
GIBSON, DUNN & CRUTCHER
333 South Grand Avenue
Los Angeles, CA 90071
(213) 229-7000

JONATHAN C. DICKEY
GIBSON, DUNN & CRUTCHER
3000 Sand Hill Road
Building 1, Suite 285
Menlo Park, CA 94025
(415) 926-7300

APPENDIX A

DESCRIPTION OF AMICI CURIAE USERS GROUPS

1. Amicus Curiae **Alamo PC Organization, Inc. [San Antonio, TX]** is a non-profit organization dedicated to furthering the knowledge of and to promote the effective use of personal computers to its members and to the public by instructional programs, activities and services. Membership consists of approximately 5,800 individual users.

2. Amicus Curiae **Buffalo IBM-PC User Group [Amherst, NY]** is a non-profit organization which provides education and information to members and the public, and provides support and assistance to members with regards to computer and related issues. Total membership is over 600, consisting almost entirely of individual computer users.

3. Amicus Curiae **Cajun Clickers Computer Club [Baton Rouge, LA]** is a non-profit organization whose purpose is to promote the education of its members and the general public in regards to computer knowledge, information and solutions to problems relative to IBM compatible computers. The group consists of approximately 1,450 individual and corporate users.

4. Amicus Curiae **Capital PC User Group, Inc. [Rockville, MD]** is a non-profit organization founded in 1982, and is one of the oldest IBM and compatible personal computer user groups in the country. It is also one of the largest, with approximately 5,800 members in the greater Washington D.C. area. The group is all volunteer, with a central theme of "users helping users." More than

60% of its members make decisions and/or recommendations about the purchase of computer equipment within their companies or government agencies.

5. Amicus Curiae **Chalmette Computer Users Group-PC [Chalmette, LA]** is a non-profit organization dedicated to the education, facilitation, and fostering of usage of computers. Total membership is 125.

6. Amicus Curiae **Channel Islands Personal Computer Users Group [Ventura, CA]** is a non-profit public benefit corporation organized for the specific purpose of educating users in the operation of personal computers, and facilitating the legitimate exchange of computer programs. Membership is approximately 400 individual users.

7. Amicus Curiae **Chicago Computer Society [Wauconda, IL]** is a non-profit organization dedicated to training and educating its members in computer-related matters. Membership is approximately 2,300 individual and corporate users.

8. Amicus Curiae **Detroit Area Network Users Group [Pleasant Ridge, MI]** is a non-profit organization dedicated to support and education for users, administrators and integrators of computer networks. The group works collaboratively with value added resellers, manufacturers, recruiters and end users in the computer network industry. Membership is approximately 300 corporate users, principally small and large users, integrators, and administrators of computer networks and related systems.

9. Amicus Curiae **Diablo Valley PC Users Group [Walnut Creek, CA]** is a non-profit organization and its principal purpose is to educate members through meetings, publications, tutorials and related activities, as well as the collection of public domain software for use by members. Membership is approximately 450 individual and corporate users.

10. Amicus Curiae **Computer Erie Bay User Group [Sandusky, OH]** is a non-profit organization dedicated to local computer users of all kinds with many levels of experience, providing support for users of various PC-compatible systems. Membership is approximately 42 individual and corporate users.

11. Amicus Curiae **Florida Association of Computer User Groups, Inc. [Winter Haven, FL]** comprises over 34 member groups, including PC and Mac groups, founded in 1993 to assist user groups in the performance of their managerial and educational activities, and is the successor organization to the Florida Sun Coast Conference of PC User Groups. Total membership of all member groups is in excess of 15,000 members.

12. Amicus Curiae **Glendora Seniors Computer Club [Glendora, CA]** is a non-profit organization for seniors fifty years of age or over, dedicated to helping members solving computer problems, presenting new or unfamiliar hardware or software, sharing ideas on how best to use existing software, and helping seniors to become better informed about computers. Membership is approximately 50 individual users.

13. Amicus Curiae **Golden Triangle PC Club [Beaumont, TX]** is the leading PC Users Group serving

southeast Texas, and strives to educate its membership and the surrounding community in the proper use of computer hardware and software for the PC platform. Membership is approximately 200 individual and corporate users.

14. Amicus Curiae **Greater Cleveland PC Users Group [Cleveland, OH]** is a group designed to enhance personal and professional growth by providing low cost education and information about current computer technology, and to provide a friendly atmosphere encouraging the exchange of ideas, information and experiences in relation to computer and information technology. The organization consists of approximately 800 individual and corporate users.

15. Amicus Curiae **IBM AB Users Group [Laguna Hills, CA]** is a non-profit organization dedicated to presenting state-of-the-art computer items to its membership. Membership is approximately 385 individual and other users.

16. Amicus Curiae **Idaho PC Users Group [Boise, ID]** is a non-profit organization which provides a means to educate and provide information of a timely nature to personal computer users, both within the membership, and among the Idaho PC user community at large. Total membership is 225, including both individual and corporate computer users.

17. Amicus Curiae **Kentucky-Indiana Personal Computer User's Group, Inc. [Louisville, KY]** is a non-profit organization which promotes the exchange of computer-related information, assists and encourages new personal computer users, and provides computer-related

services to the community. Membership is approximately 730 individual and corporate users.

18. Amicus Curiae **Long Island PC Users Group [Baldwin, NY]** is a non-profit organization which for the past fifteen years has been dedicated to disseminating information on PC products through regular meetings and seminars, formation of special interest groups for various software applications, and other activities. Membership is approximately 450 individual users.

19. Amicus Curiae **Los Angeles Computer Society [Los Angeles, CA]** is a non-profit public benefit corporation which provides a forum in the Los Angeles community for exchanging ideas and information about computers and their uses and provides education and help in solving computer-related problems. Membership is approximately 400 individual and corporate users.

20. Amicus Curiae **Macintosh User Group of Orlando [Winter Park, FL]** is a Macintosh computer interest group for the hobbyist, as well as professionals and new users, with membership of approximately 380 individual and corporate users.

21. Amicus Curiae **Manatee PC Users Group [Bradenton, FL]** is a non-profit organization designed for the educational, literary and scientific benefit of computer users. The group consists of 695 individual users.

22. Amicus Curiae **Memphis PC Users Group, Inc. [Memphis, TN]** is a non-profit organization dedicated to the education and support of all personal computer users, consisting of over 750 individual and corporate computer users.

23. Amicus Curiae **Microlink PC User Group [Whittier, CA]** is a non-profit organization consisting of approximately 70 individual users, ranging from beginners to PC industry engineers and experienced consultants. In addition to educational efforts, the group provides technical support and computer donations to charities in Southern California.

24. Amicus Curiae **Mid-Cities PC Users Group [Hurst, TX]** is an organization whose purpose is to provide a forum for members to share information and experiences that will help other members obtain maximum benefit and personal satisfaction from the use of their personal computers. The group consists of 105 individual and corporate users.

25. Amicus Curiae **Milwaukee Area IBM PC Users Group, Ltd. [Milwaukee, WI]** is a non-profit organization which provides a forum for members to share information and experiences which will help other members obtain maximum benefit and enjoyment from the use of their computers. Membership consists of approximately 150 individual and corporate users.

26. Amicus Curiae **Napa Valley PC Users Group [Napa, CA]** is a non-profit organization which provides a forum for members to share information and experiences that will help them to obtain the maximum benefit and enjoyment from the use of their computer. The group consists of 120 individual users.

27. Amicus Curiae **PC Clubhouse [Hayward, CA]** is a non-profit organization which promotes education about PC software, hardware and general usage. It has a membership of over 300 individual users.

28. Amicus Curiae **PC/Minnesota User Group, Inc. [St. Louis Park, MN]** is a non-profit organization whose mission is to utilize current and future communication technology, computer technology, and educational expertise to encourage, promote and support the exchange of information among personal computer users throughout the state of Minnesota. The organization consists of approximately 87 individual and corporate users.

29. Amicus Curiae **National Capital Tandy Computer User's Group, Inc. [Arlington, VA]** is a non-profit organization founded in 1977, at the dawn of the microcomputer revolution, to meet the needs of users of the early Radio Shack computers. Since then the organization has served members in the Washington metropolitan area, advancing the use of personal computers among members, providing a forum where personal computer information can be freely exchanged. Membership is approximately 97 individual users.

30. Amicus Curiae **PC Tech User Group [Lafayette, CA]** provides the latest technical computer knowledge to the public, and consists of over 100 individual users.

31. Amicus Curiae **Pacific Northwest PC Users Group [Bellevue, WA]** is a non-profit organization which supports the education of personal computer users through sharing knowledge and experiences, and emphasizes users helping users. Membership consists of approximately 589 individual and corporate users.

32. Amicus Curiae **Palmetto Personal Computer Club [Columbia, SC]** is a non-profit organization whose purpose is to assist DOS, Windows and OS2 type users to

better utilize their computer systems. Membership is approximately 383 individual users.

33. Amicus Curiae **Perry GA PC Users Group [Warner Robins, GA]** is a non-profit organization formed to promote the broader use, better understanding and more total application and use of programs and computers, through training and communication among members. Membership consists of approximately 50 individual users.

34. Amicus Curiae **Philadelphia Area Computer Society [Philadelphia, PA]** is a non-profit organization formed for the purpose of education and communication among computer users in the greater Philadelphia area and to inform the general public concerning computer technology and its implications for the future. Membership is approximately 1,100 individual and corporate users.

35. Amicus Curiae **Phoenix IBM-PC Users Group, Inc. [Phoenix, AZ]** is a non-profit organization which fosters an interest in IBM-type personal computers, and educates and encourages interaction between computer enthusiasts in the greater Phoenix area. The organization is comprised of approximately 900 individual and corporate users.

36. Amicus Curiae **Pioneer Valley PC Users Group [North Amherst, MA]** provides education and support for users of personal computers on home, business and educational applications. This non-profit organization has a membership in excess of 250 individual users.

37. Amicus Curiae **Polk IBM-PC Users Group, Inc. [Lakeland, FL]** is a non-profit organization which develops, encourages and stimulates interest in personal computers and their uses, and promotes the cultivation of free exchange of information among those with an interest in computer science. Membership consists of approximately 300 individual users.

38. Amicus Curiae **Quad Cities Computer Society [Davenport, IA]** is a non-profit organization whose purpose is to provide a forum for communicating ideas, resolving problems and increasing the effectiveness of users of microcomputers and related software and peripherals, and to educate the public concerning the advantages and disadvantages of microcomputers in general. Membership is approximately 800 individual and corporate users.

39. Amicus Curiae **Red River Personal Computer Club [Wichita Falls, TX]** is a non-profit organization which aids computer users in the use of computers, with an emphasis on monthly meetings, user classes and similar assistance. The group consists of approximately 125 individual users.

40. Amicus Curiae **Rockland PC Users Group [New City, NY]** is a non-profit organization, the purpose of which is to exchange computer-related ideas and information among its members, to engage in computer-related activities for the benefit of the membership, to increase the understanding and utilization of the personal computer, and to provide a medium for the exchange and propagation of public domain software, freeware and

shareware. Membership is approximately 225 individual users.

41. Amicus Curiae **Sacramento PC Users Group, Inc. [Sacramento, CA]** is a non-profit organization whose goal is to provide a forum in the greater Sacramento community for the exchange of ideas and information regarding personal computers, including IBM microcomputers and related compatible computers, and to provide assistance in solving hardware and software problems encountered during the use of these computers, among other purposes. Membership is approximately 2,600 individual users.

42. Amicus Curiae **Saginaw Valley Computer Association [Saginaw, MI]** is an organization which provides a forum for members of the IBM and compatible community for their mutual benefit, and the better utilization of their computers, and provides an opportunity for all users of IBM and compatible computers to exchange ideas, knowledge and experience. Membership is approximately 200 individual users.

43. Amicus Curiae **San Diego Computer Society [San Diego, CA]** promotes the general understanding of computers and, seeks to further the progress in the development and application of computers to our lives. The group consists of approximately 1,200 individual users.

44. Amicus Curiae **San Francisco PC Users Group [San Francisco, CA]** is a non-profit educational corporation dedicated to the collection and distribution of information concerning personal computers to its members and the general community. Membership is approximately 375 individual and corporate users.

45. Amicus Curiae **Santa Barbara PC Users Group [Santa Barbara, CA]** is a non-profit organization whose purpose is to provide information and support to users of personal computers. Membership is approximately 240 individual users.

46. Amicus Curiae **Santa Clarita Valley PC Group [Canyon Country, CA]** is an organization providing a forum for computer users to learn about PC software and hardware, and consists of approximately 145 individual users.

47. Amicus Curiae **Sierra Vista IBM PC Users Group [Sierra Vista, AZ]** is a non-profit organization which encourages the development of computer literacy and to promote the responsible use of computers. The group consists of over 50 individual and corporate users.

48. Amicus Curiae **Silicon Valley Computer Society [Santa Clara, CA]** is a non-profit organization whose primary objectives and purposes is to educate the public on personal computing through affordable classes, product demonstrations, publications, and access to technology, as well as to promote an understanding of personal computer technology and issues, and to provide a forum for public discussion of personal computing. Membership is approximately 500 individual and corporate users.

49. Amicus Curiae **South Mountain Users Group [Phoenix, AZ]** is a non-profit organization established to aid computer owners in achieving maximum results from their home or small business computers, and to serve as a reference source to make users aware of available materials. The group consists of approximately 135 individual and corporate users.

50. Amicus Curiae **Southwest Florida PC Users Group [Naples, FL]** is a non-profit organization which assists computer users in the use, enjoyment and enhancement of personal computers. Total membership is approximately 300, consisting of both individual and corporate users.

51. Amicus Curiae **Sun City Center PC Users Group [Sun City Center, FL]** is a non-profit organization of over 350 individual members, dedicated to IBM PC users, with a primary emphasis on education through classes and training on the operation of the PC.

52. Amicus Curiae **Tampa Bay Computer Society [St. Petersburg, FL]** is a non-profit organization dedicated to sharing PC knowledge, and to helping PC users learn and understand their software and hardware. Membership is approximately 2,000 individual and corporate users.

53. Amicus Curiae **Technical and Users Group Network [La Crescenta, CA]** is a non-profit organization whose specific purpose is to provide a forum for expressing ideas, resolving problems and increasing the effectiveness of users of computers, and related software and peripherals, and encouraging the creation, modification, maintenance, duplication and distribution of public domain software. Membership is approximately 200 individual and corporate users.

54. Amicus Curiae **Temple Area PC Club [Temple, TX]** is a non-profit organization dedicated to exchanging ideas and information relating to home and business computing, with membership of approximately 60 individual and corporate users.

55. Amicus Curiae **Thousand Oaks Personal Computer Club [Thousand Oaks, CA]** is a non-profit organization dedicated to enhancing the effective personal and business use of personal computers, providing feedback to the computer industry on hardware and software issues, and serving the local community as an educational resource in the personal computer field. Membership is approximately 70 individual users.

56. Amicus Curiae **Tucson Computer Society [Tucson, AZ]** is a non-profit organization of executives, entrepreneurs and home users who are interested in sharing ideas and information about the full range of personal computers. Membership is approximately 950 individual and corporate users.

57. Amicus Curiae **University of South Florida Personal Computer Users Group [Tampa, FL]** is a non-profit organization which provides education, information sharing and mutual support to computer users. The organization was founded in 1982, and consists of approximately 400 individual and corporate users, as well as faculty, staff and students of the University of South Florida.

58. Amicus Curiae **Users Personal Computer Organization [East Lansing, MI]** is a non-profit organization the purpose of which is to encourage and promote the use of personal computers and software, to assist members in enhancing their use and understanding of these systems, to provide a forum for the free exchange of information about these systems, and to take such other actions permitted by law as the membership may deem desirable. The group consists of over 50 individual and corporate users.

59. Amicus Curiae **Utah Computer Society [Salt Lake City, UT]** is organized to promote, educate and encourage the use of personal computers, with a membership in excess of 500 individual and corporate users.

60. Amicus Curiae **Utah Valley PC User Group [Provo, UT]** is a service organization consisting of approximately 125 individual and corporate users throughout Utah Valley.

61. Amicus Curiae **Warner Robins' Users Group, Inc. [Warner Robins, GA]** is a non-profit organization whose mission is to inform and help, and to educate on and about computer and software, and to provide assistance to users with computer-related problems. Total membership is 59, consisting of both individual and corporate users.

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

(23)

No. 94-2003

In The

Supreme Court of the United States

October Term, 1995

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

v.

BORLAND INTERNATIONAL, INC.,

Respondent.

On Writ Of Certiorari
To The United States Court Of Appeals
For The First Circuit

BRIEF AMICUS CURIAE OF ECONOMICS PROFESSORS AND SCHOLARS IN SUPPORT OF RESPONDENT

JOSHUA R. FLOUM
(Counsel of Record)
HELLER EHRMAN WHITE & MCAULIFFE
333 Bush Street #3100
San Francisco, California 94104-2878
(415) 772-6000

Attorneys for Amici
Economics Professors and Scholars

December 1995

COCKLE LAW BRIEF PRINTING CO., (800) 225-6964
OR CALL COLLECT (402) 342-2831

32 pp

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES iii

I. INTEREST AND IDENTITY OF THE AMICI... 1

II. SUMMARY OF ARGUMENT 1

III. ARGUMENT 3

A. Intellectual Property Policy Embodies an Economic Trade-off 3

1. Monopoly Power Is Generally Harmful.. 3

2. Intellectual Property Policy Tolerates These Harms to a Limited Extent, in Order to Reward Innovation 3

3. When (Close) Creative Alternatives Are (Close) Economic Alternatives, Copyright Does Not Confer Harmful Monopoly Power 4

B. Software Market Dynamics Can Turn Arbitrary Choices Into Compelling Choices 5

1. Network Effects 6

2. User Switching Costs 7

3. Network Effects and User Switching Costs Are Important in Software and in Spreadsheet Markets 8

4. Our Usage of the Term "Interfaces" Is Defined By Economic Properties 11

C. In the Presence of Network Effects and Switching Cost Dynamics, Copyright Protection May Confer Monopoly Power Even Absent Real Innovation 11

TABLE OF CONTENTS – Continued

Page

D. Network Effects and Switching Cost Dynamics Amplify the Harm Caused by Monopoly 13

IV. CONCLUSION 14

TABLE OF AUTHORITIES

Page(s)

CASES

Apple Computer, Inc. v. Microsoft Corp., 717 F. Supp. 1428 (N.D. Cal. 1989) 7

Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807 (1st Cir. 1995) 7, 10

MISCELLANEOUS

Kenneth Baseman, Frederick Warren-Boulton and Glenn Woroch, "The Economics of Intellectual Property Protection of Software: the Proper Role for Copyright," *StandardView* 3, June 1995 5

Luis Cabral, David Salant and Glenn Woroch, "Monopoly Pricing with Network Externalities," forthcoming in *International Journal of Industrial Organization* 13

Joseph Farrell and Carl Shapiro, "Standard Setting in High-Definition Television," *Brookings Papers on Economic Activity*, 1992 13

Joseph Farrell and Garth Saloner, Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation, *American Economic Review* 76 (1986) 12

Joseph Farrell, "Arguments for Weaker Intellectual Property Protection in Network Industries," *StandardView* 3, June 1995 13

Neil Gandal, Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities, 1994, *RAND Journal of Economics*, 25 (1994), 160-170 10

TABLE OF AUTHORITIES – Continued

Page(s)

Neil Gandal, "Competing Compatibility Standards and Network Externalities in the PC Software Market," *Review of Economics and Statistics*, November 1995, forthcoming 10

William Landes and Richard Posner, An Economic Analysis of Copyright Law, *Journal of Legal Studies*, XVIII 14

Marvin Lieberman and David Montgomery, First-Mover Advantages, *Strategic Management Journal* 9 (1988) 12

Robert Merges and Richard Nelson, On Limiting Or Encouraging Rivalry in Technical Progress: The Effect of Patent Scope Decisions, *Journal of Economic Behavior and Organization* 25 (1994) 14

Robert Merges and Richard Nelson, The Complex Economics of Patent Scope, *Columbia Law Review*, 1990 14

Michael Porter, *Competitive Strategy*, The Free Press, 1980, at 10 and 114 12

I. INTEREST AND IDENTITY OF THE AMICI

Amici are professors at Harvard University, Stanford University, Columbia University, the University of California, and other major universities, and other economic scholars who teach and write on economic issues. The economic conditions that prevail in markets such as the one before the Court in this case are subjects of intense interest and research among academic economists. Indeed, a number of amici have written scholarly papers addressing the application of intellectual property protection generally, and copyright law specifically, to "network" markets of the type at issue here. Amici do not represent either party in this action, and offer the following views on this matter in the public interest.[1] The parties have consented to the filing of this brief.

II. SUMMARY OF ARGUMENT

Intellectual property policy balances increased incentives for innovation against the harms from monopoly. Patent law confers relatively strong protection but requires evidence of significant novelty, utility, and non-obviousness. Copyright law does not demand so significant an innovation, but traditionally protects only

---

[1] None of the amici listed in Appendix A are being compensated in any way for the work on this brief. The signatories to this brief exercised complete control over its editorial contents. Respondent Borland International, Inc. helped to defray the costs of preparation of this brief. One of the primary authors of this brief has performed a small amount of consulting services to Borland in the past.

"expression" for which there exists close economic substitutes.

In computer software, however, as users invest in training, the creation of data files, and the creation of macros, all based on the interfaces of the software they are using, and as more users adopt a particular interface, what were initially arbitrary choices in the design of an interface may become compelling choices. Alternatives that were initially close creative substitutes do not remain close economic substitutes. Thus, if copyright protects an interface, it may confer substantial monopoly power and foreclose subsequent innovation.

Such a monopoly may be an appropriate reward if the entrant's product is highly innovative, and amici do not advocate denying protection to successful software products. But in the case of software interfaces, the economic dynamics may confer this reward even if the product is not highly innovative. The monopoly power results not from the superiority of the copyright holder's creation, but from the accretion of users' investments. Thus, amici believe that uncritical copyright protection for interfaces in computer software is dangerous, and on balance undesirable when these economic dynamics prevail.

III. ARGUMENT

A. Intellectual Property Policy Embodies an Economic Trade-off

1. Monopoly Power Is Generally Harmful

Economic analysis and experience alike teach that, in general, monopoly is harmful, for a variety of reasons. A single entity that controls a market may, through avarice or error, make inefficient choices; in consequence consumers suffer and have no recourse. Protected from competition, the monopoly may become wasteful. Would-be competitors and subsequent innovators may be stymied or handicapped; in the case of a legally-protected monopoly, their competitive efforts may be distorted to avoid infringing the monopoly. And – the classic pricing inefficiency of monopoly – economic value is destroyed when the seller charges prices above cost so that buyers are harmed by more than the seller benefits.

2. Intellectual Property Policy Tolerates These Harms to a Limited Extent, in Order to Reward Innovation

Such inefficiencies result even when the monopoly is granted for good reason, as is the case for much intellectual property protection. Competition in exploiting an invention would be much preferable to monopoly in exploiting the same invention, for all the above reasons, among others. But we must also take into account the incentives to invent or to create in the first place. Especially if imitation is cheap and effective, as is the case with literal copying of software "code," unrestrained

competition in exploiting a creation may greatly reduce these incentives.

Therefore, economists see intellectual property law as embodying a trade-off: it should aim to confer just enough reward to encourage desirable innovation without creating unnecessary monopoly, and should protect in ways that minimize any incidental harm caused by monopoly. Substantial monopoly power should not be granted for creations that are not significantly innovative.

3. *When (Close) Creative Alternatives Are (Close) Economic Alternatives, Copyright Does Not Confer Harmful Monopoly Power*

Patent protection requires a showing of novelty, utility and non-obviousness. Copyright protection lacks these requirements, and therefore risks conferring monopoly power where no substantially innovative contribution needs to be rewarded.

In the traditional realms of copyright, such risks have perhaps not been excessive. Since copyright traditionally protects "expressions" for which there are good alternatives, and not the underlying "idea," it does not normally confer substantial control over the entire market: Expression generally has substitutes in the traditionally creative sense, which will also be substitutes in the economic sense, limiting the copyright holder's control of economic markets.

This reassuring argument assumes that good substitutes at the time of creation remain effective substitutes and can later compete as such. It breaks down if initially

arbitrary choices, for which there are good alternatives, become economically compelling through market dynamics.

B. Software Market Dynamics Can Turn Arbitrary Choices Into Compelling Choices

The initial design choices in a computer software interface may well be largely arbitrary, in the sense that comparably good alternatives are available. However, as users acquire experience in using the product, invest in learning and in writing macros and creating files, and as more and more users adopt an interface, those alternatives may no longer be comparably good. Thus, because of the nature of users' behavior in computer software markets, initially arbitrary interface choices, for which comparably good alternatives were readily available, can become uniquely desirable for users and therefore also for competitors. For brevity we will say they become "compelling." This happens through two related economic processes that economists call "network effects" and "user switching costs."[2]

---

[2] The process by which these two economic forces combine to make copyright protection over software interfaces socially harmful is spelled out in Kenneth Baseman, Frederick Warren-Boulton and Glenn Woroch, "The Economics of Intellectual Property Protection of Software: the Proper Role for Copyright," *StandardView* 3, June 1995.

1. *Network Effects*

The English language would look highly arbitrary from the viewpoint of a "language designer" worki¬g with a clean slate. From such a viewpoint, there are a plethora of alternatives, many equally good and, surely, some better. Yet, for instance, once hundreds of millions of Americans learn the English language, these alternatives are no longer comparable. For every user, the advantages of learning and using the language of her community outweigh any benefits of superior design (such as are sometimes claimed, for example, for Esperanto). The user values, above all, the ability to communicate and "interoperate" with other users. Through the subsequent cumulation of others' choices and learning investments, the initially arbitrary language has become compelling: users will favor it even if it is not particularly good.

The same forces operate in many economic markets; economists call these forces "network effects." The classic illustration of a network effect, which is also responsible for the name, is a telephone network: the value of phone service to any individual depends on how many other individuals are connected. More generally, a network effect is an economic force that makes a product more valuable to each user, the more other users own that product or a sufficiently compatible one. In addition to direct network effects such as the telephone example, "indirect" network effects may be created through greater supply of complementary products.

Network effects are important in software markets.[3] Users want to share data files and programs such as macros; they want to work on machines owned by others; they want access to a wide selection of complementary products (including third-party manuals, consulting services, training courses, and add-on software). Certain aspects of programs must be identical in order for users of different programs to share these network benefits; these aspects will predictably include "interfaces" and aspects of a program that define a language, such as a macro language.[4]

2. *User Switching Costs*

Software users invest in complementary products and services, and in creating files and programs. Consider two programs that incorporate different arbitrary choices in their interfaces – choices that are equally good from the users' point of view prior to purchase and use. Because of these different interface specifications, the user's investments in learning, file creation, *etc.* are difficult or costly to transfer from one program to the other. Even if the user would be indifferent between the two interfaces *ex ante*, once she has invested in one system, she will substantially prefer it because she would have to

---

[3] The presence of network effects was accepted by the court for personal computer operating system software in *Apple Computer, Inc. v. Microsoft Corp.*, 717 F. Supp. 1428, 1431 (N.D. Cal. 1989).

[4] Amicus Brief of Computer Scientists Re Copyrightability of Computer Languages, *Lotus Development Corporation v. Borland International, Inc.*, December 1993, at 2 (J.A. 10).

replicate these investments in order to switch to the other.[5]

Because of users' reluctance to switch, alternative interfaces that were equally good at the time of initial design do not remain equal and are not equal from the point of view of software designers who must choose an interface specification at a later date. Consequently, a new program attempting to compete with an established program will find that the first one's initially arbitrary interface choices have become economically compelling.

3. *Network Effects and User Switching Costs Are Important in Software and in Spreadsheet Markets*

Network effects and user switching costs are very strong in the computer software industry. Developers are keenly aware of the importance of compatibility to users. They take pains to ensure that new versions of a software package are compatible with earlier versions, both to minimize their own customers' switching costs and to

[5] As an analogy, consider the potential for introducing competition in local telephone service. One issue in this area is whether a subscriber should have the right to keep her telephone number if she switches to a competing carrier. The user has "invested" by telling her friends her number and printing it on checks and stationery, for instance. If the local (currently regulated) monopoly telephone company had "copyright" over subscribers' telephone numbers, she would be much more reluctant to switch to a competing supplier. Each person's number was initially arbitrary as far as she was concerned, but is now valuable to her.

maximize their network effects. This lesson was convincingly driven home to Lotus when it launched the Release 2.0 version of its *1-2-3* spreadsheet that was not fully compatible with the previous 1A version.[6] Compatibility at the user interface is also crucial since users do not have to learn a new set of keystrokes nor to "unlearn" keystrokes which have become automatic. This is one reason that Borland's programs (and most other popular spreadsheet and word processing packages) often provide a "chameleon interface" that allows users to choose one familiar to them and avoid learning a new one, inefficiently replicating their investments.

In the case of spreadsheet software, it clearly became compelling for competitors to offer compatibility with Lotus *1-2-3*.[7] Econometric evidence confirms that users

[6] "Compatibility was at the top of the list [of product design issues]. We actually had an experience around compatibility with Release 2.0 which was not totally nourishing for [Lotus], and it was after the release of Release 2.0 and the subsequent release of 2.01 where the importance of compatibility was firmly stamped on everyone's forehead as the single, unifying concept that we had to manage from generation to generation of our products." Deposition of Jim P. Manzi, Chief Executive Officer, Lotus Development Corp., August 22, 1991, p. 174, J.A. 715.

[7] It was important for *Quattro* to be compatible with files and macros created with *1-2-3* so that users would not have to replicate their investments. The Appeals Court fully appreciated the value of compatibility to users: "Under the district court's holding, if the user wrote a macro to shorten the time needed to perform a certain operation in Lotus 1-2-3, the user would be unable to use that macro to shorten the time needed to perform that same operation in another program. Rather, the user would have to rewrite his or her macro using that other

express their preference for compatibility in their spreadsheet purchase decisions.[8]

To illustrate network effects and their impact on competition, consider a computer user who wishes to buy a spreadsheet program and has narrowed his choice to two competing spreadsheet products. The first product will give him access to a large selection of spreadsheet data files created by others using that product's data format: for instance, many publicly available government records use Lotus' data format. The second spreadsheet product has no such installed base, but has certain superior features desired by the prospective new user. He would prefer to buy the second product, but only if it can read the pre-existing data files which are based on the first product's format – in other words, only if it is compatible to that extent.[9]

---

program's menu command hierarchy. This is despite the fact that the macro is clearly the user's own work product." *Lotus Development Corp. v. Borland International, Inc.*, 49 F.3d 807, 818 (1st Cir. 1995), Pet. App. at 20a.

[8] Neil Gandal, "Hedonic Price Indexes for Spreadsheets and an Empirical Test of the Network Externalities," *RAND Journal of Economics*, 25 (1994), 160-170 finds that "consumers are willing to pay a significant premium for spreadsheets that are compatible with the Lotus platform."

[9] Neil Gandal, "Competing Compatibility Standards and Network Externalities in the PC Software Market," *Review of Economics and Statistics*, November 1995, forthcoming, discusses these effects empirically in spreadsheet and database markets.

4. *Our Usage of the Term "Interfaces" Is Defined By Economic Properties*

Amici do not claim to know exactly which aspects of computer programs must be compatible in order for the programs to remain as competitive after the build-up of network effects and switching costs as they would be *ex ante*. In general, interface specifications must be compatible for this to occur. For convenience, therefore, we use the term "interface aspects" to mean those choices.

C. In the Presence of Network Effects and Switching Cost Dynamics, Copyright Protection May Confer Monopoly Power Even Absent Real Innovation

Whether or not network effects and switching costs limit competition depends crucially on whether or not vendors have proprietary control of the interfaces. If interfaces are public, competitors can make their products compatible, and users will be able to choose a program on the basis of its quality and price rather than on switching costs and installed base of users. If interfaces are protected by copyright, the copyright holder can prevent competitors from making their products compatible. In this way the intellectual property treatment of interfaces crucially affects the nature of competition.

If an established seller controls network effects, a competitor must either convince users that a new product will succeed broadly (as distinct from simply appealing to a particular user), or else persuade them that, despite the disadvantage in network effects, the product

improvement is so dramatic that they should switch anyway. Similarly, if existing users must bear switching costs (must replicate their private investments) in order to buy from a competitor, the competitor operates at a disadvantage. It is well recognized in economics and in competitive strategy that these effects provide an advantage and an opportunity for long-term profits to an incumbent, even absent any inherent superiority of its product.[10]

In this way the established product, even if not highly innovative, may acquire substantial monopoly control through the copyright protection of its interfaces. Such a result is contrary to what we understand to be the usual pattern of copyright, which provides narrow protection by protecting only "expression" and refusing to protect an "idea," and to the usual pattern of patent law, which confers potentially broad control as a reward for demonstrably innovative contributions. Those usual patterns make economic sense in terms of the intellectual-property tradeoff described above, unlike the outcome when copyright confers broad control with no showing of innovativeness.[11]

---

[10] Michael Porter, *Competitive Strategy*, The Free Press, 1980, at 10 and 114 (and elsewhere); Marvin Lieberman and David Montgomery, "First- Mover Advantages," *Strategic Management Journal* 9 (1988), 41-58; Joseph Farrell and Garth Saloner, "Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation," *American Economic Review* 76 (1986), 940-955.

[11] When copyright protection confers large rewards on interface creations that are not particularly innovative, two further dynamic inefficiencies arise. Firms may deliberately create incompatible interfaces in the hope of being the lucky focus of network effects, but meanwhile the market is inefficiently splintered among incompatible interfaces. And firms may race to introduce products prematurely – an incentive that may be partly responsible for the prevalence of "bugs" in new software releases.

D. Network Effects and Switching Cost Dynamics Amplify the Harm Caused by Monopoly

Monopoly power is likely to be particularly harmful in markets in which network effects and user switching costs are important. As with any monopoly, above-cost pricing will deter purchases by many potential users who value the product more than it costs to produce. Economic efficiency is thereby harmed in any market. But when network effects are important, there is an additional effect: those who do buy get a less valuable product as a result of the smaller network. Thus, where network effects are present, the ordinary pricing inefficiency of monopoly is likely to be amplified.[12]

---

[12] See, for instance, Joseph Farrell and Carl Shapiro, "Standard Setting in High-Definition Television" *Brookings Papers on Economic Activity*, 1992, at 41-42, and Joseph Farrell, "Arguments for Weaker Intellectual Property Protection in Network Industries," *StandardView* 3, June 1995, 46-49. In principle the monopoly right holder could adopt "penetration pricing" and thus reward early purchasers for the benefits they provide to later users. Under this scheme prices start low and become higher as the product becomes established. More often, the tendency to "price skim" overwhelms any attempt at penetration pricing. The "price skimming" strategy is to set prices high initially to extract profits from users who value the product highly, and then gradually reduce price over time to make sales to other users. See Luis Cabral, David Salant and Glenn Woroch, "Monopoly Pricing with Network Externalities," forthcoming in *International Journal of Industrial Organization*.

The pricing inefficiency of monopoly is not the only concern, however. In computer software, cumulative innovation is important – developers of the next generation of software products benefit from the breakthroughs, and try to avoid the pitfalls, of their predecessors. Consequently, it is highly desirable that all comers be able to build on the existing state of knowledge.[13] In cumulative innovation markets, overly strong intellectual property protection may actually retard rather than encourage innovation; thus there may not be any social benefits from increased incentives for innovation to weigh against the social costs of monopoly distortions.[14]

IV. CONCLUSION

For the reasons given above, amici believe that economic efficiency argues strongly against *uncritical* protection of *interface* aspects of computer software. It is economically harmful to protect, through copyright's uncritical mechanism, an aspect of computer software that is initially arbitrary but then becomes compelling. As

---

[13] *See* Brief Amicus Curiae of Computer Scientists, Section III B 2.

[14] William Landes and Richard Posner, "An Economic Analysis of Copyright Law," *Journal of Legal Studies*, XVIII, June 1989, 325, 348, argue that overprotection will reduce the number of products. Robert Merges and Richard Nelson, "The Complex Economics of Patent Scope," *Columbia Law Review*, 1990, and "On Limiting Or Encouraging Rivalry in Technical Progress: The Effect of Patent Scope Decisions," *Journal of Economic Behavior and Organization* 25 (1994), 1-24, describe how strong intellectual property protection has retarded cumulative innovation in a number of industries.

copyright law does not generally protect compelling "choices," amici urge the Court to find that the *initial* or *ex ante* arbitrariness of interface design means that it should not be granted broad protection. Rather, only those aspects of a software program whose value, if any, stems from their originality and quality should be protected. Those aspects whose potential value will be due to network effects or user switching costs should not be. On amici's understanding of the facts in this case, when Borland introduced *Quattro*, it undertook the extra effort to make its product compatible largely in order to avoid imposing switching costs on Lotus users who might switch to *Quattro* and to be on an equal footing in respect of network effects.

Intellectual property protection should reward software developers for their innovative contributions. For economic efficiency, these returns should encourage innovators to create software products in number and quality so as to maximize the overall well-being of users and creators jointly. This almost certainly involves a prohibition on literal copying of code. It may also involve protection of broader aspects of a software product, especially if the creation is in fact highly innovative (and we express no view here on whether Lotus' interfaces were). This may well involve large rewards for highly innovative products and we certainly do not advocate punishing winners. But protection of software should not *uncritically* protect aspects that confer substantial monopoly power over a significant market segment, as amici believe will be the case if interfaces such as program

menu commands are automatically protected by copyright.

Respectfully submitted,

JOSHUA R. FLOUM
(Counsel of Record)
HELLER EHRMAN WHITE &
McAULIFFE
333 Bush Street #3100
San Francisco, California
94104-2878
(415) 772-6000

Attorneys for Amici
Economics Professors and Scholars

December 8, 1995

APPENDIX A
LIST OF SIGNATORIES TO THE BRIEF

[Primary Authors]

1. Joseph Farrell
Department of Economics
University of California, Berkeley

Joseph Farrell is Professor of Economics and Affiliate Professor of Business at the University of California, Berkeley. His teaching has included graduate level classes in industrial organization, microeconomics, and competitive strategy. He is (jointly) North American Editor of the Journal of Industrial Economics, and Associate Editor of The Economics of Innovation and New Technology. He has been a reviewer for the National Science Foundation, the National Academy of Sciences, and the Office of Technology Assessment on matters involving innovation and high technology. He has been a consultant for the US Department of Justice and for private firms in computer software and related industries. He has been a leader in developing the economic theory of compatibility standards and of user switching costs, and has published numerous academic articles on these subjects.

2. Glenn Woroch
Haas School of Business
University of California, Berkeley

Professor Woroch is presently Lecturer and Director, Consortium for Research on Telecommunications Policy, Haas School of Business at the University of California, Berkeley. He received an M.A. in Statistics and a Ph.D. in Economics from Berkeley. He teaches industrial organization, regulation and microeconomics at Berkeley, and he has

previously taught economics at the University of Rochester and at Stanford University. He was also a research economist at GTE Laboratories. He is currently a member of the editorial boards of Information Economics & Policy and The Journal of Regulatory Economics, and a member of the board of directors of the International Telecommunications Society. He has consulted for the Departments of Energy and Commerce on economic and regulatory issues and regularly serves on industry committees. Besides publishing numerous articles on regulation, antitrust and industrial organization, he has recently published research on the economics of intellectual property protection of computer software.

[Additional Signatories – In alphabetical order]

3. Kenneth J. Arrow
Joan Kenney Professor of Economics Emeritus and Professor of Operations Research Emeritus
Departments of Economics and Operations Research
Stanford University

Professor Arrow has also been a member of the faculty at the University of Chicago and Harvard University and a Research Associate of the Cowles Commission for Research in Economics. He received his Ph.D. in Economics from Columbia University in 1951. He has published papers and books in the fields of social choice, general economic equilibrium, medical care, and the economics of information and innovation. He has been president of several learned societies and has received several honorary awards, including the Nobel Memorial Prize in Economic Science in 1972.

Statement: I agree fully with the general principles of this brief, especially the need to insist on a high degree of novelty for recognizing copyright

protection in interfaces. I am not acquainted with the specific facts in this case and take no stand about any of them.

4. W. Brian Arthur

Professor Arthur is Morrison Professor of Economics at Stanford and Citibank Professor at the Santa Fe Institute. He has written numerous articles and a book (Increasing Returns and Path Dependence in the Economy, University of Michigan Press, 1994) on the dynamics of network effects. He is a Fellow of the Econometric Society and was awarded the Schumpeter Prize in 1990.

5. Sanford V. Berg, Ph.D.
Department of Economics
University of Florida

Sanford V. Berg, Ph.D., is Distinguished Service Professor in the Department of Economics at the University of Florida. He is also the Florida Public Utilities Professor and the Director of the Public Utility Research Center where he organizes conferences and workshops on regulatory issues. In addition, he has served as a consultant to various private and public organizations including state regulatory commissions, corporations, the Congressional Office of Technology Assessment, and the World Bank. Presently, he is a project co-director of the Telecommunications Industry Analysis Project (TIAP). He was selected as the University of Florida Outstanding Undergraduate Teacher of the Year (1993). In 1994, he was the University's nominee for the Professor of the Year Award (Carnegie Foundation for the Advancement of Teaching). He has published widely on business and economics topics, including copyright protection, compatibility standards, and

the determinants of innovative activity. He is the co-author of several books, including *Natural Monopoly Regulation: Principles and Practice* (Cambridge University Press, 1989). Dr. Berg graduated from the University of Washington with honors in Economics, and received his Ph.D. degree from Yale University in 1970.

6. Jay Pil Choi
Department of Economics
Columbia University

Jay Pil Choi is Assistant Professor of Economics at Columbia University. He received his Ph.D. at Harvard in 1990 and was a Post-Doctoral Fellow at Tilburg University, the Netherlands. He is also a recipient of the prestigious Abe Fellowship. He has published numerous articles on intellectual property rights and R&D competition.

7. Paul A. David

Senior Research Fellow, All Souls College, Oxford University, and Professor of Economics, Stanford University.

Visiting Research Professor of Science and Technology, Rijksuniversitiet Limburg, Maastricht.

Quondam William Robertson Coe Professor of American Economic History, and former chair of Economics Department, Stanford University. Current Program Leader of Stanford University Center for Economic Policy Research's High Technology Impact Program.

Fellow of American Academy of Arts and Sciences, Fellow of the British Academy, Fellow of

International Econometric Society. Former President of the Economic History Association.

Author of numerous articles and books on technology and economic growth, including recent studies of the economics of information networks, inter-operability standards, and the evolution of intellectual property protection regimes.

Consultant to National Academy of Sciences, National Science and Engineering Council of Canada, the World Bank, the European Commission, the OECD, etc.

8. Nicholas Economides

Professor of Economics at the Stern School of Business of New York University

His fields of specialization and research include industrial organization, the economics of networks, especially of telecommunications and of information, the economics of technical compatibility and standardization, and the structure and organization of financial markets. He has published widely in the areas of networks, telecommunications, oligopoly, positioning of differentiated products, and on liquidity and the organization of financial markets and exchanges. He holds a Ph.D. and a M.A. in Economics from the University of California at Berkeley as well as a B.Sc. (First Class Honors) in Mathematical Economics from the London School of Economics. He has previously taught at Columbia University (1981-1988) and at Stanford University (1988-1990). He is editor of the International Journal of Industrial Organization, and associate editor of Journal of Regional Science. He is currently editing a special issue of the International Journal of Industrial Organization on Network Economics. His book,

"Communications Convergence: Economic Perspectives on Quality and Market Evolution," jointly written with Bob Dansby, is forthcoming from MIT Press and the American Enterprise Institute.

9. Aaron Edlin
Department of Economics
University of California, Berkeley

Professor Edlin is on the faculty of the economics department of the University of California at Berkeley. He received his A.B. in the science and policy program of the Woodrow Wilson School at Princeton University, and his Ph.D. in economics and J.D. in law from Stanford University. In research, he specializes in industrial organization, law and economics, and public economics. He teaches industrial organization and public economics.

10. Neil Gandal
Assistant Professor of Economics
Tel Aviv University

Professor Gandal received his Ph.D. in Economics from the University of California-Berkeley in 1989. One of his areas of expertise is the Economics of Compatibility and Standardization. He has published or co-authored a number of the papers in this area, including:

"Competing Compatibility Standards and Network Externalities in the PC Software Market," *Review of Economics and Statistics*, November 1995.

"Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities," 1994, *RAND Journal of Economics*, 25: 160-170.

11. Jerry Richard Green
Department of Economics
Harvard University

Jerry Green is the John Leverett Professor in the University and the David A. Wells Professor of Political Economy in the Department of Economics. He joined the Harvard faculty in 1970, chaired the Economics Department from 1984 to 1987, and served as Provost of the University from 1992 to 1994.

Professor Green is a Fellow of the Econometric Society and served on its Council from 1988 to 1994. He is a Fellow of the American Academy of Arts and Sciences and has been an Erskine Fellow at the University of Canterbury, and a Guggenheim Fellow. He is an Oversees Fellow of Churchill College, Cambridge University. In 1980, he received the J.K.Galbraith Prize for excellence in teaching.

Professor Green chaired the National Science Foundation's Information Sciences Advisory Panel in 1980, prepared the Foundation's Ten-Year Outlook for the Social Sciences in 1983 and served on the National Academy of Sciences Panel on Taxpayer Compliance in 1984. He is a member of the Board of Trustees of the Beth Israel Hospital of Boston where he serves on the budget and finance committee and chairs the committee on conflict of interest policy. He has been an advisor to many universities and foundations.

Professor Green is known for his work on the theories of incentives, rational expectations, and behavior under uncertainty. He has contributed to a number of areas in applied economics, including tax policy, finance, health economics, higher education, and patent policy. He is the author of Incentives in Public Decision Making (with Jean-Jacques Laffont, 1978), Microeconomic Theory (with Andreu Mas-

Colell and Michael Whinston, 1995) and over eighty scientific articles.

12. Michael D. Intriligator
Professor of Economics, Political Science and Policy Studies
University of California, Los Angeles

Michael D. Intriligator is Professor of Economics at the University of California, Los Angeles (UCLA), where he is also Professor of Political Science, Professor of Policy Studies in the School of Public Policy and Social Research, and Director of the Jacob Marschak Interdisciplinary Colloquium on Mathematics in the Behavioral Sciences. From 1982 to 1992 he was Director of the UCLA Center for International and Strategic Affairs. He has been a member of the UCLA faculty since 1963, the year he received his Ph.D. in Economics at MIT working under Robert M. Solow. He teaches courses in economic theory, econometrics, mathematical economics, and international relations. His papers in economics are in the areas of economic theory, econometrics, health economics, and the transition to a market economy. He has published more than 100 journal articles and other publications and is the author of *Mathematical Optimization and Economic Theory* (Prentice-Hall, 1971) and *Econometric Models, Techniques, and Applications* (Prentice-Hall 1978, second edition with Ronald G. Bodkin and Cheng Hsiao, forthcoming, 1995). He is a Fellow of the Econometric Society, a member of the Council on Foreign Relations (New York) and the International Institute for Strategic Studies (London), and is listed in *Who's Who in America* and *Who's Who in Economics*.

13. Louis Kaplow
Harvard Law School

Louis Kaplow is a Professor of Law at Harvard Law School and a Research Associate at the National Bureau of Economic Research. He received his J.D. and Ph.D. (in economics) both from Harvard University. His teaching and research involve the application of economic analysis to law, with a specialization in competition policy. Among his writings in this area are an article on the relationship between intellectual property and competition policy and the text, Antitrust Analysis, co-authored with Phillip Areeda. Professor Kaplow also serves on the editorial board of a number of journals and as a referee for many of the leading economics journals that publish articles in this field. In addition, he has advised the Department of Justice and testified before Congress on issues pertaining to the appropriate relationship between intellectual property and competition policy.

14. Suzanne Scotchmer
Professor of Economics and Public Policy
Graduate School of Public Policy
University of California, Berkeley

Professor Scotchmer was previously Associate Professor of Economics at Harvard University, and has been Visiting Professor at the University of Paris I, University of Toronto, and New School of Economics, Academy of Sciences, Moscow, Russia. In the last several years she has held research fellowships at Yale University (Olin Fellow) and Stanford University (Hoover National Fellow). She received her M.A. in Statistics from U.C., Berkeley in 1979 and her Ph.D. in economics from U.C. Berkeley in 1980. She has published widely on the economics of intellectual property law, economics of crime with

an emphasis on rules of evidence in the law, economics of tax evasion, cooperative game theory, club theory, and evolutionary game theory. Her research in the past has been supported mostly by the National Science Foundation, by also the Sloan Foundation, the Olin Foundation, the Harris Trust, and Phi Beta Kappa.

Among her editorial activities, she has served on the editorial board of two of the three journals of the American Economic Association, the *American Economic Review* and *Journal of Economic Perspectives*, and also on the editorial boards of the *Journal of Public Economics* (published at the London School of Economics) and *Regional Science and Urban Economics* (a North-Holland Journal). She referees for all the major economics and policy journals.

15. Frederick R. Warren-Boulton
MiCRA
1875 I Street, N.W.
Washington, DC 20006

Frederick R. Warren-Boulton is a principal of MiCRA, a Washington-based economics consulting and research firm. He received a B.A. degree from Yale University, an M.P.A. from the Woodrow Wilson School of Princeton University, and a Ph.D. in Economics from Princeton University. From 1972 to 1983 he was an Assistant and then Associate Professor of Economics at Washington University in St. Louis, where he taught the graduate level courses in microeconomics and industrial organization. From 1983 to 1989, he served as the chief economist for the Antitrust Division of the U.S. Department of Justice, first as Director of its Economic Policy Office and then as Deputy Assistant Attorney General for Economic Analysis.

Since leaving the government, he has served as a Resident Scholar at the American Enterprise Institute, a Visiting Lecturer of Public and International Affairs at the Woodrow Wilson School at Princeton University, and Research Associate Professor of Psychology at The American University. He has published extensively in the areas of antitrust, regulation and intellectual property, and has served as an expert witness or consultant on a number of cases involving antitrust or intellectual property issues in the computer industry.

MOTION FILED
DEC 1 [redacted]

(24)

No. 94-2003

In the

Supreme Court of the United States

October Term, 1995

Lotus Development Corporation,

Petitioner,

v.

Borland International, Inc.,

Respondent.

ON WRIT OF CERTIORARI TO THE
UNITED STATES COURT OF APPEALS
FOR THE FIRST CIRCUIT

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE AND BRIEF AMICUS CURIAE OF SOFTWARE PROTECTION COMMITTEE OF MINNESOTA INTELLECTUAL PROPERTY LAW ASSOCIATION IN SUPPORT OF RESPONDENT

Steven W. Lundberg*
Bradley A. Forrest
Daniel J. Kluth
Schwegman, Lundberg & Woessner, P.A.
80 S. Eighth Street, #3500
Minneapolis, MN 55402
(612) 373-6902

Rudolph P. Hofmann Jr.
Westman, Champlin, Kelly, P.A.
121 S. Eighth Street, #720
Minneapolis, MN 55402
(612) 334-3222

Nelson R. Capes
Palmatier, Sjoquist & Helget, P.A.
7900 Xerxes Ave. S., #2000
Minneapolis, MN 55431
(612) 831-5454

Counsel for Amicus Curiae

**Counsel of Record*

1995 – Bachman Legal Printing ☎ (612) 339-9613 ⊠ FAX (612) 337-9613

BEST AVAILABLE COPY

31pp

IN THE
SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE

The Software Protection Committee of the Minnesota Intellectual Property Law Association, Inc., 514 Nicollet Mall, Suite 300, Minneapolis, MN 55401, ("MIPLA"), a regional association of primarily attorneys, whose interest and practices lie in the areas of copyright, patent, trademark, trade secret and other intellectual property law, respectfully moves for leave to file the attached brief *amicus curiae* in this copyright infringement case. The consent of the attorney for respondent has been obtained. Consent of the attorney for

the petitioner had not been obtained at the time this motion was printed.

The Software Protection Committee of MIPLA has over forty members and offers the views of a leading regional group of intellectual property attorneys concerned with copyrights and patents for software, representing a broad range of views. The Software Protection Committee of MIPLA therefore respectfully requests that the Court grant its motion for leave to file the accompanying brief, and consider the Software Protection Committee's arguments during its deliberations.

The attached brief was not reviewed by MIPLA as a whole, and represents only the views of the Software Protection Committee of MIPLA.

Respectfully submitted,

STEVEN W. LUNDBERG*
CHAIR OF AD HOC SUBCOMMITTEE OF
SOFTWARE PROTECTION COMMITTEE
MINNESOTA INTELLECTUAL
PROPERTY LAW ASSOCIATION, INC.
3500 IDS Center
80 South Eighth Street
Minneapolis, MN 55402
(612) 373-6902
NELSON R. CAPES, BRADLEY A. FORREST,
RUDY P. HOFMANN, JR. & DANIEL J. KLUTH
Counsel for Amicus Curiae

December 4, 1995
* Counsel of Record

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES iii

INTEREST OF AMICUS CURIAE 1

SUMMARY OF ARGUMENT 2

ARGUMENT 3

I. SECTION 102(B) EXCLUDES ELECTRONIC OPERATION FROM COPYRIGHT PROTECTION 3

II. COMPUTER PROGRAMS AND ELECTRONIC OPERATION ARE SEPARATE AND DISTINCT 7

III. BORLAND IS FREE TO PROVIDE A COMPUTER HAVING AN IDENTICAL ELECTRONIC OPERATION AS LOTUS 1-2-3 10

IV. THE LOTUS 1-2-3 MENU COMMAND HIERARCHY CHARACTERIZES ELECTRONIC OPERATION, NOT A PROTECTIBLE ASPECT OF THE LITERARY FORM OF A COMPUTER PROGRAM 11

V. SOFTWARE COPYRIGHTS VS. SOFTWARE PATENTS: RELATIVE RISK AND RETURN 16

A. LENGTH OF PROTECTION 16

B. RIGHTS PROTECTED 16

C. COST AND DIFFICULTY OF OBTAINING PROTECTION 17

D. FRINGE BENEFITS 18

E. COPYRIGHT FOR *DE FACTO* STANDARDS 20

VI. CONCLUSION 22

TABLE OF AUTHORITIES

Cases **Page**

Apple Computer, Inc. v. Microsoft Corp, No. 93-16883 (9th Cir. Sept. 19, 1994) 11
Ashton-Tate Corp. v. Ross, 728 F.Supp 597 (D. N. Cal. 1989), *aff'g* 916 F.2d 516 (1990) 11
Baker v. Selden, 101 U.S. 99 (1879) 5
Digital Communications Assoc. v. Softklone Distrib. Corp., 659 F.Supp 449 (D. N. Ga. 1987) 11
Dred Scott v. Sandford, 60 U.S. (19 How.) 393, 564 (1857) (Curtis, J., dissenting) 7
Feist Publications Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991) 3
Lemelson v. United States, 752 F.2d 1538, 224 U.S.P.Q. (BNA) 526 (Fed. Cir. 1985) 17
Lotus Dev. Corp. v. Borland Int'l Inc., 49 F.3d 807 (1st Cir. 1995) 10, 13
Mazer v. Stein, 347 U.S. 201 (1954) 5
NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d (BNA) 1177 (1989) 9, 11
Pennwalt Corp. v. Durand-Wayland, Inc., 833 F.2d 931, 3 U.S.P.Q.2d (BNA) 1737 (Fed. Cir. 1987) (en banc) . 17
Philbrook v. Glodgett, 421 U.S. 707 (1975) 4

Constitutional Provisions, Statutes and Public Laws

17 U.S.C. § 101 5
17 U.S.C. § 102 18
17 U.S.C. § 102(b) 2, 3, 4, 5, 6, 7, 9
17 U.S.C. § 302 16
35 U.S.C. § 101 18
35 U.S.C. § 102 18
35 U.S.C. § 103 18
35 U.S.C. § 112 ..... 17, 20
35 U.S.C. § 154 16

37 C.F.R. § 1.20(c) 19
37 C.F.R. § 1.20(e),(f),(g) 17
37 C.F.R. § 1.510 19
37 C.F.R. § 202.20(c)(vii)(2) 20
37 C.F.R. § 202.3(c) 18
Pub. L. No. 93-573, § 201(b)-(c),
88 Stat. 1973-74 (1974) 4

Legislative Materials

H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976) *reprinted in*
1976 U.S. Code Cong. and Admin. News 5659 4

Other Authorities

Gene K. Landry, *Macros and Broad Software Copyright Protection: The Latest Round in* Lotus v. Borland, Computer Law., Vol. 11, No. 2, Feb. 1994 ..... 21, 22
Steven W. Lundberg, George H. Gates III, and John P. Sumner, *Baker v. Selden, Computer Programs, 17 U.S.C. § 102(b) and Whelan Revisited* 13 Hamline L. Rev. 221 (1990) 6, 9
Gregory J. Maier, *Software Protection - Integrating Patent, Copyright and Trade Secret Law*, 69 J. Patent and Trademark Off. Soc'y 151 (1987) 6
Final Report of the Nat'l Comm. On Technological Uses of Copyrighted Works (1978) 4
M. Nimmer, 3 Nimmer on Copyright § 13.01[B] 20
Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copyright Protection For Commercialized University Research*, 42 Vand. L. Rev. 639 (1989) 5
E. Sergant, *Justice Touched with Fire*, *in* Mr. Justice Holmes 206-07 (F. Frankfurter ed. 1931) 7

Frederick R. Warren-Boulton, et. al., *POINT: Copyright Protection of Software Can Make Economic Sense*, Computer Law., Vol. 12, No. 2, Feb. 1995 ..... 21, 22

IN THE
SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

---

BRIEF AMICUS CURIAE OF SOFTWARE PROTECTION COMMITTEE OF THE MINNESOTA INTELLECTUAL PROPERTY LAW ASSOCIATION, INC. IN SUPPORT OF RESPONDENT

INTEREST OF AMICUS CURIAE

The Software Protection Committee of the Minnesota Intellectual Property Law Association, Inc., 514 Nicollet Mall, Suite 300, Minneapolis, MN 55401, ("MIPLA") files this *amicus curiae* brief to aid the Court in its review of the ruling of the United States Court of Appeals for the First Circuit in the above-titled case.

The Software Protection Committee of MIPLA offers the views of a leading regional group of intellectual property attorneys concerned with copyrights and patents for software, representing a broad range of views. The Software Protection Committee of MIPLA has over forty members, many of them with extensive practical experience in applying copyright and patent law to protect computer software. Attorney members of Software Protection Committee represent both copyright owners and their competitors. Accordingly, the Software Protection Committee's opinion is the product of a non-partisan analysis. This brief has not been reviewed by MIPLA as a whole, and represents only the views of the Software Protection Committee.

MIPLA is a regional association of primarily attorneys, whose interest and practices lie in the areas of copyright, patent, trademark and other intellectual property law. Membership in MIPLA is drawn from private law firms, corporations and universities.

SUMMARY OF ARGUMENT

The Copyright Act of 1976 expressly excludes from copyright protection any idea, procedure, process, system, method of operation, concept, principle or discovery.[1] A computer program and the electronic operation of a computer executing the program are separate and distinct. While copyright clearly extends to a computer program as a literary work, Congress intended that electronic operation be excluded from protection under Section 102(b). Characteristics of electronic operation can be separated from the characteristics of the literary form of a computer program. The former can be excluded from protection under Section 102(b) without directly impinging on the protection of the latter.

The menu command hierarchy Lotus seeks to protect specifies how an electronic spreadsheet device will respond to a set

[1] 17 U.S.C. § 102(b).

of electronic symbols[2] when applied to the device in a defined syntax. This hierarchy constitutes choices of electronic design which were antecedent to the coding of the Lotus 1-2-3 computer program, and are separate and distinct therefrom. Moreover, the hierarchy, disembodied from a specific literary form, characterizes the operation of an electronic device or computer and not the literary form of a computer program or user-interface, and as such is uncopyrightable. To the extent a computer program or user interface must embody a "copy" of the electronic symbols and/or syntax logic of the menu command hierarchy in order to achieve the electronic operation defined thereby, then such commands and/or syntax are necessary to the electronic operation of, or part of a method of operating, a computer, and are therefore uncopyrightable. The First Circuit was correct, therefore, in finding that the menu command hierarchy of Lotus 1-2-3 was uncopyrightable under 17 U.S.C. § 102(b).

ARGUMENT

I. Section 102(b) Excludes Electronic Operation from Copyright Protection

The Copyright Act of 1976 expressly excludes from copyright protection any idea, procedure, process, system, method of operation, concept, principle or discovery. 17 U.S.C. § 102(b). Section 102(b) identifies those elements of a work, otherwise deserving of copyright protection, for which copyright is not available. *Feist Publications Inc. v. Rural Telephone Serv. Co.*, 499 U.S. 340, 356 (1991). Additionally, the legislative history of Section 102(b) makes it clear that Congress intended to exclude the

[2] These digital symbols are translatable to English language letters which have a verbal relationship to the electronic operation invoked by the symbols when supplied to a computer. The letters themselves, apart from their associated digital symbols, have no meaning to the computer.

electronic operation[3] of a computer, as opposed to the mode of expression, from software copyright protection.[4] In particular, the House Report on the 1976 Copyright Act sets forth:

> Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied within the program are not within the scope of copyright law.

H.R. Rep. No. 1476, 94th. Cong., 2d Sess. 57 (1976) *reprinted in* 1976 U.S.C.C.A.N. 5659, 5660.

The idea/expression dichotomy with respect to software was also given treatment in the 1978 Report of the National Commission on New Technology Uses of Copyrighted Works (CONTU) preceding the 1980 amendment to the Copyright Act.[5] Following extensive study, CONTU sought to clarify the distinction between copyrightable computer programs and uncopyrightable electronic operation when it stated that copyright law protects the program as long as it remains fixed in a tangible medium of expression, but not the electro-mechanical functioning of a machine. *See, Final Report of the Nat'l Comm. On Technological Uses of Copyrighted Works*, at 48-49 (1978). Consequently, the definition of "computer program" as recommended by CONTU and

---

[3] As used herein, the term "electronic operation" refers to the electro-mechanical functioning of a computer and to electronic systems, methods, or processes carried out on a computer.

[4] Legislative intent governs decisions on the construction of statutes. *E.g., Philbrook v. Glodgett*, 421 U.S. 707, 713 (1975).

[5] Congress established CONTU in part to study the relationship between copyright protection and computer technology, and to make recommendations to amend the copyright law where appropriate. Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1973-74 (1974).

set forth in the 1980 amendment to the Copyright Act is simply "a set of statements or instructions to be used directly or indirectly in a computer to bring about a certain result". 17 U.S.C § 101.

The Supreme Court's first comprehensive review of the idea/expression dichotomy occurred in *Baker v. Selden*, 101 U.S. 99 (1879), later restated in *Mazer v. Stein*, 347 U.S. 201, 217 (1954), which established the principles subsequently codified in § 102(b). The Court in *Baker* held that the plaintiff's copyright of a textbook describing a new method of accounting did not grant plaintiff a monopoly on the accounting system and that "writings" necessary to use the accounting system were *per se* uncopyrightable. *Id.* at 103. The Court's analysis turned on the proposition that copyright in a writing should not be used to indirectly obtain protection to utilitarian works more properly the subject of patent protection.[6] To merely state that *Baker*, and subsequently § 102(b), asserts a generalized idea/expression dichotomy is to gut the meaning therein. The idea/expression dichotomy existed since the earliest origin of copyright law, and well before *Baker*.[7]

The Court in *Baker* did not apply an idea/expression analysis of the traditional type. Rather, the Court in *Baker* permitted the defendant's reproduction of materials otherwise copyrightable under a traditional idea/expression analysis, in order to facilitate the use by the defendant of the unprotectible utilitarian features of the accounting system. *Id.* at 104. The *Baker* Court also held that the explanatory essay was protected by copyright, but not infringed. *Id.* Viewed as such, the analysis under *Baker*, and its corresponding provision § 102(b), requires defining the unprotected utilitarian feature, and then determining whether

---

[6] *Id.* at 103.

[7] *See*, Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copyright Protection For Commercialized University Research*, 42 Vand. L. Rev. 639, 693 n.288 (1989).

copyright protection for the corresponding writing would affect the right of others to practice the feature.

In the context of computer programs, the statutory mandate requires drawing a distinction between the electronic operations performed by a computer and the set of statements or instructions of a computer program.[8,9] In other words, Congress limited the definition of copyright protection available to a computer program to its literary form, i.e. its "statements and instructions", to exclude the electronic operation of a computer from copyright protection. Thus, to the extent that electronic operation can be excluded from protection while maintaining protection for the literary form, the Court has received a statutory mandate to do so.

In the past, the plain meaning of section 102(b) has been ignored by many courts, contrary to the intent of Congress, presumably to fill in the gaps left when courts denied patents for software inventions. *See*, Gregory J. Maier, *Software Protection - Integrating Patent, Copyright and Trade Secret Law*, 69 J. Patent and Trademark Off. Soc'y 151, 161 (!987). Stretching copyright protection in order to protect a programmer's valuable efforts may be understandable or even equitable, but if it is not within the confines of the statutory mandate, it is not acceptable.[10] The

[8] *See* Steven W. Lundberg, George H. Gates III, and John P. Sumner, *Baker v. Selden, Computer Programs, 17 U.S.C. § 102(b) and Whelan Revisited* 13 Hamline L. Rev. 221, 228 (1990).

[9] To help keep the distinction between electronic operation and computer programs clear within this brief, the term "computer program" is only used to refer to the literary form of a computer program, and not to the electronic operation effectuated by the computer program when it is executed in a computer.

[10] When a strict interpretation of a statute, according to the fixed rules which govern the interpretation of laws, is abandoned, and the theoretical opinions of individuals are allowed to control its meaning, we no longer have the statute; we are under the government of individuals, who for the time being have power to declare what the statute is, according to their own personal views of what it ought to mean. *See*, *Dred Scott v. Sandford*, 60 U.S. (19 How.) 393, 564 (1857) (Curtis, J., dissenting).

decisions ignoring or misapplying the idea/expression dichotomy intended by Congress proceed from sympathy rather than law.[11] As such, Section 102(b) instructs the Court to cover only the literary form of a computer program, and to exclude from protection the electronic operation of computers.

II. Computer Programs and Electronic Operation are Separate and Distinct

As demonstrated below, the literary form of a computer program is separable from the electronic operation resulting from the execution of the program in a computer. Characteristics of literary form, as that term is used herein, include the literal sequence of program code,[12] the structure, sequence and organization of the code, and literary or audiovisual elements of a user-interface resulting from the execution of the code in a computer.

Characteristics of the literary form of program code do not include characteristics of electronic operation resulting from the execution of the code in a computer. Characteristics of electronic operation include, *inter alia*, electronic functions[13] or processes for transforming data from one state to another, the electronic flow of

[11] It is reported that Justice Holmes and Judge Learned Hand once had lunch together and afterward, as Justice Holmes began to leave in his carriage, Judge Hand ran after him shouting, "Do justice sir, do justice!" Justice Holmes stopped the carriage and rebuked Justice Hand: "That is not my job. It is my job to apply the law." E. Sergant, *Justice Touched with Fire*, *in* Mr. Justice Holmes 206-07 (F. Frankfurter ed. 1931).

[12] As used herein, the term "code" refers to the "statements or instructions" of a computer program.

[13] Such as moving a column of numbers in a spreadsheet, or storing data in a file.

data through a computer, and the control or invoking of electronic functions, processes, or data flow by commands or data supplied to the computer by a user or external device. The lexicon of electronic characteristics is independent of whether the electronic function is "hard-wired" (*i.e.*, no software) or implemented with a programmable device.

Electronic operation or functionality is separable from the literary form of program code when a multiplicity of computer programs with dissimilar literary forms[14] can be used to achieve the desired electronic operation or functionality.[15] In such cases, the literary form of any one of these dissimilar computer programs can be granted generous copyright protection without precluding others from attaining the same electronic operation using dissimilar code.

The electronic operation resulting from a given computer program, when executed in a computer, can be defined by electronic characteristics. If there exists, for the purpose of obtaining electronic operation having the defined electronic characteristics, a multiplicity of computer programs of dissimilar literary form which are plausible alternatives to one another, and if the defined characteristics do not intrinsically suggest one alternative form over the other, then it can be said that the defined characteristics do not provide "access"[16] to the given computer

[14] Dissimilar literary forms are those that are not substantially similar, that is they do not use the same literal sequences of code, nor the same structure, sequence or organization of code.

[15] In cases of merger of literary form with electronic operation, none of the literary forms are protectible because there are few or only one form capable of effectuating the desired electronic function.

[16] This term is used in its sense as an element of proof of copyright infringement.

program, and are separable therefrom.[17] This principle has even been applied to separate the characteristics of electronic operation from the literary form of machine code,[18] and can be used for user-interfaces.[19]

A defined set of electronic characteristics which do not provide access to the literary form of an associated computer program can be said to be non-informing.[20] Any set of non-informing electronic characteristics can therefore be treated as *per se* uncopyrightable under 17 U.S.C. § 102(b) without impinging on copyright protection for the literary form of a computer program. These non-informing characteristics belong to the universe of processes, systems and/or methods of operation, as those terms were intended by Congress to apply to exclude protection to the "electro-mechanical functioning of a machine."

It follows from the above that infringement of a computer program cannot be established on the basis of copying non-informing electronic characteristics alone. In all cases, infringement must be based on the copying of the literary form of

[17] An example of a "non-informing" definition of electronic characteristics is the user manual of Lotus 1-2-3. This manual describes the operation of the Lotus 1-2-3 electronic spreadsheet but, with the exception of the description of the literary form of the user-interface, does not describe in any way the literary form of the computer program Lotus uses to obtain this operation.

[18] *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d (BNA) 1177 (1989). Machine code has a very close correspondence to electronic operation.

[19] It is noted that by definition a user-interface provides access and control over the electronic operation obtained by the associated computer program. There is therefore a reasonable likelihood that many elements of a user-interface will be necessary to use of or part of the method of operation of such uncopyrightable electronic operation, and hence uncopyrightable under 17 U.S.C. § 102(b).

[20] For an example of this principle, see Steven W. Lundberg, George H. Gates III, and John P. Sumner, *Baker v. Selden, Computer Programs, 17 U.S.C. § 102(b) and Whelan Revisited* 13 Hamline L. Rev. 221, 22-230 (1990).

a computer program, and not on similarity of electronic operation. This distinction, fundamental to the proper application of copyright law to computer software, has been lost on a number if not most courts.[21] In the instant case, the First Circuit properly recognizes the distinction between a computer program and electronic operation.[22]

Because the above-recited principles can be used to cleanly reverse-engineer uncopyrightable electronic operation from the protectible literary form of a computer program, electronic operation can be freely excluded from copyright protection without direct effect on the protection of the literary form of computer programs.[23]

III. Borland is Free to Provide a Computer Having an Identical Electronic Operation as Lotus 1-2-3.

Under current copyright law in all the Circuits, provided one does not copy the literal code, the look and feel, the sequence, structure and organization, nor the filtered protectible elements of the Lotus 1-2-3 computer program, one is free to duplicate the electronic operation realized by running the program on a suitable

[21] This is most evident in the uninformed manner in which many prior decisions on copyright protection for software use the term computer program in two different senses: in one sense the term is used to refer to an inert literary work, and in another sense to the electronic operation resulting from the program when it is executed in a computer. These uninformed mixed uses of this term have muddled the distinction between literary form and electronic operation.

[22] The First Circuit refers to the electronic form simply as "Lotus 1-2-3," and does not confuse a computer program with electronic operation. *Lotus Dev. Corp. v. Borland Int'l Inc.*, 49 F.3d 807, 809 (1st Cir. 1995).

[23] Of course, the literary form of a computer program will be subject to other exclusions from copyright, such as uncopyrightable ideas, *scenes a faire* and merger.

computer.[24] This electronic operation can include the exact same electronic functions as Lotus 1-2-3 without worry of copyright infringement. Many cases have so held under the idea versus expression analysis.[25]

IV. The Lotus 1-2-3 Menu Command Hierarchy Characterizes Electronic Operation, not a Protectible Aspect of the Literary Form of a Computer Program

The menu command hierarchy of the Lotus 1-2-3 spreadsheet system at issue in this case is a command set that has a defined syntax or logical system relating the commands to one another for the purpose of controlling an electronic spreadsheet system. Each of the commands, when used as part of a macro language, take the form of unique electronic binary symbols.[26]

[24] Computers may provide electronic function with no programming at all. In fact, most of the first computers were special purpose machines designed to provide electronic operations such as the adding of two numbers. Software, as we know it today had not yet been invented. Electrical circuitry and logic gates provided the ability to accept numbers, modify them and provide an output. Software was invented to make the computers easy to modify to perform new tasks or electronic operations. It is very possible to provide a hardware-only machine to implement an electronic spreadsheet that appears to operate identical to the Lotus 1-2-3 electronic spreadsheet system running on a programmable computer. Copyright cannot be used to prevent making such a machine and should not be used to prevent use of it.

[25] *E.g., see NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d (BNA) 1177 (1989), *supra, Apple Computer, Inc. v. Microsoft Corp*, No. 93-16883 (9th Cir. Sept. 19, 1994), *Ashton-Tate Corp. v. Ross*, 728 F.Supp. 597 (D.N. Ca. 1989), *aff'g* 916 F.2d 516 (1990), *Digital Communications Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F.Supp. 449 (D. N. Ga. 1987).

[26] Actual literary forms of these commands exist only in the form of a visualization of the menu command hierarchy in the user interface of the Lotus 1-2-3 electronic spreadsheet system, in a displayed or printed macro file, or in the documentation for the system. Each of the binary symbols are chosen from a set of electronically encoded ASCII alpha-numeric characters (one symbol for each character of the alphabet and the numbers 1-10, as well as for many other

These binary symbols, used in certain combinations as prescribed by the rules of the syntax, are provided to a computer or other electronic device to invoke electronic operations such as copying a column or row of a displayed spreadsheet.[27]

In order to provide electronic operation in accordance with the commands and syntax of the menu command hierarchy of the Lotus 1-2-3 spreadsheet system, an electronic device must recognize these commands and the syntax of their use. If the electronic device is hard-wired (i.e. contains no software), the circuits are designed to recognize the binary symbols of the commands and respond to them in accordance with the syntax, which is also encoded in the form of electronic logic.[28] Copyright cannot be used to prevent making such a machine and should not be used to prevent use of it. If the electronic device is a programmable computer, the electronic binary symbols and the syntax logic are embodied in a computer program used to obtain the desired electronic operation.

Whether one implements the menu command hierarchy of the Lotus 1-2-3 electronic spreadsheet system in a hard-wired device or a programmable one, the disembodied[29] menu command hierarchy is nothing but a specification of uncopyrightable

---

common punctuation marks or abbreviations), so that the symbol also represents a letter of the alphabet that has a verbal relationship to the command.

[27] In essence, as stated by the First Circuit, the commands are like electronic buttons that are part of the method of operating the Lotus 1-2-3 electronic spreadsheet.

[28] This is not only theoretically possible, it is technically quite easy to achieve.

[29] "Disembodied" is used here in the sense of being divorced from any particular embodiment in the literary form of a computer program.

electronic operation.[30] Furthermore, the disembodied symbols and syntax logic of the Lotus 1-2-3 electronic spreadsheet system constitute choices of electronic design which were antecedent to the coding of the Lotus 1-2-3 computer program,[31] and were intended from the start to constitute characteristics of electronic operation.

The central point is that Lotus is not entitled to copyright a set of electronic symbols and syntax logic that they have made necessary to the operation of an electronic device.[32] The Court should not be confused by the fact that these symbols also represent letters of the English alphabet (according to the ASCII standard) that describe or suggest the electronic operation invoked by the symbols. In point of fact, these letters themselves have no meaning to a computer; what Lotus really seeks is to prevent Borland from programming a computer to map the electronic symbols and syntax logic of the Lotus 1-2-3 menu command hierarchy to the corresponding electronic operation of Borland's spreadsheet.

To the extent the Lotus 1-2-3 computer program embodies a "copy" of the electronic symbols and/or syntax logic of the hierarchy, the symbols and their syntax logic constitute a preexisting work merely incorporated in the program to provide a

---

[30] It is particularly plain to see that the syntax constitutes electronic characteristics, as it defines how a computer will respond to different combinations of inputs, has no value or use outside of defining this response, and because it is not readily seen how this syntax is embodied in literary form in a computer program.

[31] *See Lotus Dev. Corp. v. Borland Int'l Inc.*, 49 F.3d 807, 816 (1st Cir. 1995) (Speaking of the underlying computer code being different from the menu command hierarchy).

[32] If Lotus can do this, then the electronic symbols used for internal messaging and control by other kinds of electronic devices also should be copyrightable, provided they are chosen from a set of ASCII symbols having the pretense of a literary work. This would, in effect, extend copyright protection to all digital electronics.

new work of authorship constituting a compilation or collection of this preexisting work and the code of the computer program.[33] As such, the commands and their syntax cannot correctly be characterized as non-literal aspects of the program,[34] even if both are covered by the same copyright registration.

In fact, if the disembodied menu commands and their syntax *per se*[35] are not considered part of the Lotus 1-2-3 computer program, it is clear that the disembodied menu command hierarchy which Lotus seeks to protect is both separable from and non-informing with respect to the literary form of the code of the Lotus 1-2-3 computer program.[36] As such, as established in Section III above, Lotus cannot maintain that the code of its Lotus 1-2-3 computer program has been infringed on the basis of copying the disembodied menu command hierarchy.

To establish a case of infringement of the Lotus 1-2-3 computer program, Lotus must at a minimum show that Borland copied some aspect of the manner in which the menu command

---

[33] Even if the menu command hierarchy was not "fixed" prior to the Lotus 1-2-3 computer program, it would still be separable therefrom as electronic characteristics. This case presents the somewhat unusual case where the codes used for electronic operation also correspond by translation to a "classic" literary form.

[34] That these commands and syntax are embodied in some literary form in the Lotus 1-2-3 computer program does not transform them from uncopyrightable electronic characteristics to copyrightable non-literary elements of a computer program. By that reasoning otherwise uncopyrightable telephone listings could be made copyrightable simply by incorporating them in a computer program.

[35] *I.e.*, not of any particular literary form in the program.

[36] More specifically, there is no way to know from this disembodied hierarchy what literary form a computer program will take for the purpose of embodying the hierarchy. There are a multitude of different ways the commands and syntax of the Lotus 1-2-3 spreadsheet system can be embodied in a computer program.

hierarchy is embodied in the Lotus 1-2-3 computer program or the user-interface of the Lotus 1-2-3 spreadsheet system. Here, Lotus has done neither.

As pointed out above, the disembodied menu command hierarchy of the Lotus 1-2-3 spreadsheet system, standing alone, clearly constitutes uncopyrightable characteristics of the electronic operation. Furthermore, to the extent it is necessary to embody some literary form of the menu command hierarchy into a computer program to obtain the uncopyrightable electronic operation defined in the hierarchy, then the embodiment of that literary form is necessary to the use of the electronic operation,[37] and therefore unobjectionable under copyright law.[38,39] Accordingly, the First Circuit was correct to exclude the menu

---

[37] Or, as put by the First Circuit, the menu command hierarchy is a part of the method of operating the uncopyrightable spreadsheet system.

[38] It is noted again that Lotus has not alleged that Borland has copied any particular literary form of the menu command hierarchy, but only that it has been necessarily copied in some form, and that they are entitled to protect all forms, however embodied, under copyright. It is noted that the visualization or literary form of the user interface, and the program code itself of the Lotus 1-2-3 computer program may be copyrightable as literary works provided they may be expressed in many ways, but they are not the subject of this appeal. The question before this court is whether the disembodied menu command hierarchy, which when implemented on a computer and allows a user to control the Lotus 1-2-3 electronic spreadsheet system, is copyrightable in and of itself.

[39] As Borland has implemented commands in the hierarchy, a user does not see anything of the user interface that users of a computer running Lotus 1-2-3 see. Rather, the commands are translated or interpreted in real time into commands that are usable by a computer running the Borland Quattro program. A table of user input commands, comprising the first letter of the user commands is used to make that translation. When a command is received, the computer searches down the table to identify the correct Quattro code to execute. No organization of the independent commands was copied from Lotus' organization of the code corresponding to such commands other than that necessarily incident to implement sets of hierarchically related commands, such as move a row or column.

command hierarchy of Lotus 1-2-3 from copyright protection.

V. Software Copyrights vs. Software Patents: Relative Risk and Return

The treatment of software copyrights by some courts, such as the *Lotus* district court, result in a larger universe of protected rights at significantly lower cost than would result from a corresponding United States patent. This does not appear to be sound policy, nor in keeping with Congress' intent, as noted above, to limit the scope of copyright protection for software. Set forth below is a comparison of patent and copyright protection for computer programs.

A. Length of Protection

Patents on applications filed after June 8, 1995, have a term of protection that begins on the date of grant and ends twenty years from the filing date of the application for the patent.[40] Copyright in a work created after January 1, 1978 generally endures for a term consisting of the life of the author plus fifty years.[41] Clearly, a software copyright has a term much longer than a corresponding software patent.

B. Rights Protected

A United States patent grants to the patentee, his heirs or assigns, the right to exclude others from making, using, or selling the invention throughout the United States.[42] The "invention"

[40] 35 U.S.C. § 154.

[41] 17 U.S.C. § 302.

[42] 35 U.S.C. § 154.

refers to the invention literally set forth in the claims[43] plus substantial equivalents.

In order to infringe a patent, a device must include each and every limitation literally set forth in at least one patent claim, or the substantial equivalent thereof.[44,45] Patent claims thus provide a very good guide to the extent of protection provided by the patent. If even one claimed element is missing from an accused device, the accused device will not literally infringe the patent.[46]

In contrast, a copyright registration application has no such property definition. There is thus little certainty for a competitor who wishes to develop a competitive product: until the copyright owner brings suit for infringement, the developer cannot know which aspect of the computer program, i.e., source code, object code, structure, sequence and organization, user interface, or parameter lists the copyright owner may assert as infringed.

C. Cost and Difficulty of Obtaining Protection.

A United States patent typically costs the patentee between $5,000.00 to $25,000.00 to obtain. Thereafter, at three, seven, and eleven years after issue, maintenance fees of $990.00, $1990.00, and $2990.00 are due.[47] An application for copyright application

[43] 35 U.S.C. § 112.

[44] *Lemelson v. United States*, 752 F.2d 1538, 224 U.S.P.Q. (BNA) 526 (Fed. Cir. 1985).

[45] *Pennwalt Corp. v. Durand-Wayland, Inc.*, 833 F.2d 931, 3 U.S.P.Q.2d (BNA) 1737 (Fed. Cir. 1987) (en banc).

[46] Thus, the claims of a software patent perform the very important function of notifying competitors of what they may and may not make, sell and use without infringement.

[47] 37 C.F.R. § 1.20(e),(f),(g).

for a single work must be accompanied by a fee of $20.00.[48] No other administrative or maintenance fees are required during the term of the copyright.

The difficulty of obtaining a U.S. patent is also enormously greater than of obtaining a copyright registration. To obtain a U.S. patent, a patentee must show that the invention is novel,[49] non-obvious,[50] and useful.[51] The novelty and non-obvious barriers are difficult to surmount and generally require extensive prosecution by the patent attorney.[52]

In contrast, the copyright statute only requires "original" works of authorship.[53] There is no requirement for novelty, non-obviousness, or utility. In particular, a copyright author need not demonstrate that he was the first to create the particular expression embodied in the work.

D. Fringe Benefits

Not only is a software copyright cheaper to obtain in relation to the extent of rights protected than a software patent, but a software copyright owner also enjoys a number of additional

---

[48] 37 C.F.R. § 202.3(c).

[49] 35 U.S.C. § 102.

[50] 35 U.S.C. § 103.

[51] 35 U.S.C. § 101 ("Whoever invents or discovers any new and useful. . .").

[52] In addition, for software inventions, there is the "mathematical algorithm" bar to patentability. This is a court-developed (and PTO-developed) doctrine that has generally prevented the grant of a patent to software inventions which are mere mathematical algorithms on the grounds that they do not meet statutory grounds for patentability under 35 U.S.C. § 101. This doctrine is a formidable barrier to patentability of software inventions.

[53] 17 U.S.C. § 102.

benefits, many of them relating to infringement, that a patent owner does not enjoy.

First, a software copyright is much less susceptible to a challenge of invalidity than a software patent. It is a virtual certainty that if a software patentee sues an alleged infringer, the alleged infringer will counterclaim that the patent is invalid, either on the grounds that it is anticipated or obvious over the prior art, or that "inequitable conduct" was used in procuring the patent. Although an alleged copyright infringer may also challenge the copyright on the grounds that it does not contain copyrightable subject matter, the standard for copyrightability is originality, which is much less difficult to uphold than the novelty standard of patentablility. The copyright owner thus has substantially less risk in attempting to enforce his exclusive rights than does the patentee.

Furthermore, a competitor need not manufacture a competitive device and take a chance on being sued for patent infringement. For a fee, the competitor may request that the Patent and Trademark Office re-examine the issued patent on the basis of prior art patents or printed publications which the competitor believes invalidate the patent.[54] There is no similar mechanism available to a competitor concerned about a software copyright: he must essentially produce the competing product and take his chances in court. Thus, the owner of the software copyright has much more surety of maintaining his monopoly.

The proofs required for copyright infringement are a mixed blessing for the copyright owner. He is disadvantaged relative to the patentee in that the copyright owner must prove that the alleged infringer copied the work, while the patentee need not do so (innocent infringement is not a defense in a patent infringement suit). However, copying may be proved by circumstantial evidence, usually on the basis that the alleged infringer had access

---

[54] 37 C.F.R. § 1.510; 37 C.F.R. § 1.20(c).

to the allegedly copied work.[55] In software copyright infringement, especially for mass-distributed software, this would not appear to be a difficult problem.

On the other hand, the patentee must, in order to prove literal infringement, show that each and every element and limitation of at least one patent claim is present in the accused device. A competitor can sometimes readily "design around" a patent by making slight modifications.

Finally, the software copyright owner, but not the patentee, may maintain important aspects of his work as trade secrets while still enjoying a statutory monopoly. The Copyright Act allows the copyright owner to block out significant portions of the work deposited with the Copyright Office.[56] In contrast, the patentee must disclose the "best mode" contemplated by the inventor of carrying out his invention.[57] If the patentee attempts to keep the best mode a trade secret, his patent may later be held invalid.

In summary, the software copyright owner arguably enjoys a larger universe of protected rights, with additional fringe benefits, at enormously lower cost of procurement than the software patentee.

E. Copyright for *De Facto* Standards

A recent article presents a very forceful economic argument that copyright protection of *de facto* standards should not be

[55] M. Nimmer, 3 Nimmer on Copyright § 13.01[B].

[56] 37 C.F.R. § 202.20(c)(vii)(2).

[57] 35 U.S.C. § 112.

granted.[58] The article argues that granting copyright protection to *de facto* standards limits the "network externalities" which occur "when the value of a product or service to a buyer increases with the cumulative number of other buyers".[59]

Decisions like *Lotus v. Borland* make it less likely that the network benefits of standard user interfaces will be realized. Lotus 1-2-3 users, who have invested thousands of dollars in training on the standard 1-2-3 user interface, will have only one source for a spreadsheet program that uses this standard interface, i.e., Lotus 1-2-3. As is described in the Landry article,[60] the "slippery slope" of this decision may lead to a "broadening of copyright protection for program command protocols without logical limit."[61] This may have enormous consequences in the areas of program-to-program communication, modems, telecommunications, and computer languages.[62]

Furthermore, the problem of software copyright protection of non-literal program elements extends well beyond *de facto* standards, as has been discussed above. The point is not that Lotus, or any developer of a computer program, should be denied all protection for that program, but that the scope of protection awarded should be commensurate with the cost of obtaining that protection and competitors should have clear notice, *ex ante*, of those aspects which are being protected.

[58] Frederick R. Warren-Boulton, et. al., *POINT: Copyright Protection of Software Can Make Economic Sense*, Computer Law., Vol. 12, No. 2, Feb. 1995 (hereinafter "Warren-Boulton").

[59] *Id.* The example given is that of the telephone network.

[60] Gene K. Landry, *Macros and Broad Software Copyright Protection: The Latest Round in Lotus v. Borland*, Computer Law., Vol. 11, No. 2, Feb. 1994.

[61] Landry, *supra*, at n.39.

[62] *Id.*

Warren-Boulton et. al. argue that:

> the uncritical, unexamined, and extensive nature of protection under copyright is efficient only when the expected welfare loss of its being granted mistakenly is *de minimis*. Like summary judgment and the *per se* rule, copyright is the cost-effective approach only when the probability of error (i.e., overprotection) is very small, and/or when the potential social loss associated with the error is very small.[63]

A significant probability of overprotection may exist anytime copyright protection is given to non-literal program elements in the realm of electronic operation. This is the case because extending copyright protection to these elements results in greater protection than afforded by a patent at enormously lower cost.

VI. Conclusion

In the computer program copyright paradigm Lotus seeks the Court to adopt, electronic operation is indistinct from and in fact merged with the literary form of a computer program. Thus, according to Lotus, it is not possible to separate protectible writing from electronic operation, and precluding protection to the latter *per se* implicitly violates the intent of Congress to protect the former. This is not the case and Lotus' position should be rejected.

In its holding, the First Circuit found, in essence, that the menu command hierarchy was a part of the electronic operation of Lotus 1-2-3, and was therefore uncopyrightable. The First Circuit's decision respects the paradigm sought by Congress in which electronic operation is excluded from copyright protection,

---

[63] *Id.*

while at the same time permitting liberal copyright protection to the literary form of computer programs. Accordingly, the decision should be upheld.

Respectfully submitted,

STEVEN W. LUNDBERG (Counsel of Record)
CHAIR OF AD HOC SUBCOMMITTEE OF
SOFTWARE PROTECTION COMMITTEE
MINNESOTA INTELLECTUAL
PROPERTY LAW ASSOCIATION, INC.
3500 IDS Center
80 South Eighth Street
Minneapolis, MN 55402
(612) 373-6902
NELSON R. CAPES, BRADLEY A. FORREST,
RUDY P. HOFMANN, JR. & DANIEL J. KLUTH
Counsel for Amicus Curiae
December 4, 1995